(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

ANALYSIS OF A DATA COMMUNICATION NETWORK'S PERFORMANCE UNDER VARYING RETRANSMISSION DISCIPLINES

by

John R. Kirwan, Jr.

September, 1990

Thesis Advisor: Donald P. Gaver

Approved for public release; distribution is unlimited

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority 2b Declassification/Downgrading Schedule	3 Distribution/Availability of Report Approved for public release; distribution is unlimited.
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School — 6b Office Symbol *(if applicable)* 55	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization — 8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers Program Element No / Project No / Task No / Work Unit Accession No

11 Title *(include security classification)* ANALYSIS OF A DATA COMMUNICATION NETWORK'S PERFORMANCE UNDER VARYING RETRANSMISSION DISCIPLINES

12 Personal Author(s) John R. Kirwan Jr.

13a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
Master's Thesis	From To	September 1990	137

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	DDN, network simulation, data communication network

19 Abstract *(continue on reverse if necessary and identify by block number)*

A stochastic simulation model is developed, using the SLAM II simulation language, to study the dynamics and performance of a small data communication network. The simulation program models pertinent aspects of Defense Data Network (DDN) protocols. The effect of changes in node-to-node and host-to-host retransmission timeout intervals upon expected response time is studied using the model.

20 Distribution/Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified
22a Name of Responsible Individual D. P. Gaver	22b Telephone *(include Area code)* (408) 646-2605 — 22c Office Symbol 55Gv

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Analysis of a Data Communication Network's Performance Under Varying Retransmission Disciplines

by

John R. Kirwan Jr.
Commander, United States Navy
B.S., U.S. Naval Academy, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1990

Author: ______________________
John R. Kirwan Jr.

Approved by: ______________________
D. P. Gaver, Thesis Advisor

____
P. A. Jacobs, Second Reader

____
Peter Purdue, Chairman,
Department of Operations Research

ABSTRACT

A stochastic simulation model is developed, using the SLAM II simulation language, to study the dynamics and performance of a small data communication network. The simulation program models pertinent aspects of Defense Data Network (DDN) protocols. The effect of changes in node-to-node and host-to-host retransmission time-out intervals upon expected response time is studied using the model.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. PURPOSE

The purpose of this thesis is to develop a simulation model of a data communication network which, while simplified, will capture pertinent aspects of Defense Data Network protocols. This simulation model is then used to study aspects of network performance. The study specifically includes the analysis of network performance during operations at various node-to-node and host-to-host retransmission timeout intervals. The measure of performance emphasized here is response time.

B. SUMMARY

A stochastic model for a packet switching network resembling the Defense Data Network (DDN) is written in the simulation language SLAM II. This simulation models a simplified data communication network topology. It implements a network flow and logic such that the underlying protocols utilized within the Defense Data Network are modeled. Experiments are then conducted to study the effect of operation of the network under 1) different host-to-host retransmission timeout intervals and 2) different node-to-node retransmission timeout intervals. The measure of performance adopted is the *mean response time*, i.e. the time required for a message, consisting of a group of packets, to be transmitted from one host computer to a second host computer; conflicting traffic present in the network affects that transmission time. Since a finite number of simulations can be run under given conditions all performance measures are statistical estimates and, hence, subject to sampling error.

II. THE DEFENSE DATA NETWORK (DDN)

A. GENERAL DESCRIPTION

The Defense Data Network (DDN) is a packet switching network implemented to meet the Department of Defense data communication requirements. The two functional areas of the DDN are the network backbone and subscriber access. The network backbone is made up of the packet switching nodes (PSN) and interswitch trunk (IST) circuits. The subscriber access area is made up of the circuits, interface equipment, and compatible protocols to permit access to the backbone network. [Ref. 1: p 1-1]

1. Evolution of the DDN

The DDN is an evolutionary outgrowth of the Advanced Research Projects Agency Network (ARPANET) which is an experimental network chartered to advance the state-of-the-art in computer networking. ARPANET originally supported both Department of Defense (DOD) and non-DOD users. As DOD requirements grew, it became clear that there was a need for a DOD exclusive communication network that incorporated precedence handling and security features. This requirement resulted in the development of the Military Network (MILNET). Further expansion lead to the development of secure networks which were incorporated into the DDN. The present components of the DDN are the unclassified MILNET, the secret level DSNET1, the top secret level DSNET2, and the sensitive compartmented information level DSNET3. [Ref. 1: pp 2-1,2-2]

The current "mission of the DDN is to supply common-user data communications backbone services in support of military operational systems, to include intelligence systems, command and control systems, general purpose ADP and other long-haul data communication networks." [Ref. 1: p 2-2]

2. DDN Architecture

The architecture of a network is concerned with the way in which various functions of a network interact. The most common network architecture is to use layering of functions. A layer or level is the hierarchical organization of network functions so that network design complexity is reduced. Each layer performs its functions in a modular fashion, interfacing only with the layers immediately above and below it. In this manner a lower layer provides services to a higher layer while shielding it from the details of the operations of the lower layer(s). During device-to-device communications layer

n of device A communicates with layer n of device B through the conventions defined by the n^{th} level protocols. Actual data is only transferred between devices at the lowest layer, the physical layer. [Ref. 2: pp 10-11]

The backbone network of DDN is made up to the four lowest layers as defined by the International Standards Organization (ISO) Open Systems Interconnection (OSI) model. These are the physical, where raw data bits are transferred over communication links; the data link, where packet framing and acknowledgement occurs; the network, where subnet control such as routing and host interface occurs; and transport, where host-to-host functions take place. [Ref. 2: pp 15-17]

Figure 1 on page 4 shows a comparison of the OSI layers to the corresponding DDN protocols. [Ref. 3: p 4-18a]

3. Functional Areas

The backbone network is made up of the packet switching nodes (PSN) and the interswitch trunks (IST) which connect PSN-to-PSN. PSN's serve as the interface between subscriber hosts and the backbone network as well as serving as the forwarding point for packets being communicated over the network. IST's are high speed links that carry data between PSN's utilizing common-carrier circuits, military microwave, military satellites, and commercial satellites. The PSN's operate at the ISO level 1-3 layers; the IST's are a layer 1 function. [Ref. 1: p 3-1,3-3]

Subscriber access can be accomplished by direct communication to a PSN (if the host is configured with appropriate software and hardware) or through an access device (which provides required interfaces). Subscriber access incorporates the implementation of a set of protocols (conventions) including mandatory use of the Internet Protocol (IP), the Transmission Control Protocol (TCP), and either the X.25 or ARPANET Host Interface Protocol, all of which are described later. [Ref. 1: p 4-1]

B. DEFENSE DATA NETWORK PROTOCOLS AND FUNCTIONS

1. DDN Protocols

The protocols in use within DDN establish general, minimum specifications (a convention) for any hardware/software combination to be connected to the DDN. By providing these conventions for each function of the network, the designers of the DDN have provided the latitude so that various common-user networks, using different hardwares/softwares, may connect to the backbone network with no modification to that network's upper level functions. The protocols are described in great depth in military standards and packet switching node specifications. [See references 4,5.] The fol-

n of device A communicates with layer n of device B through the conventions defined by the n^{th} level protocols. Actual data is only transferred between devices at the lowest layer, the physical layer. [Ref. 2: pp 10-11]

The backbone network of DDN is made up to the four lowest layers as defined by the International Standards Organization (ISO) Open Systems Interconnection (OSI) model. These are the physical, where raw data bits are transferred over communication links; the data link, where packet framing and acknowledgement occurs; the network, where subnet control such as routing and host interface occurs; and transport, where host-to-host functions take place. [Ref. 2: pp 15-17]

Figure 1 on page 4 shows a comparison of the OSI layers to the corresponding DDN protocols. [Ref. 3: p 4-18a]

3. Functional Areas

The backbone network is made up of the packet switching nodes (PSN) and the interswitch trunks (IST) which connect PSN-to-PSN. PSN's serve as the interface between subscriber hosts and the backbone network as well as serving as the forwarding point for packets being communicated over the network. IST's are high speed links that carry data between PSN's utilizing common-carrier circuits, military microwave, military satellites, and commercial satellites. The PSN's operate at the ISO level 1-3 layers; the IST's are a layer 1 function. [Ref. 1: p 3-1,3-3]

Subscriber access can be accomplished by direct communication to a PSN (if the host is configured with appropriate software and hardware) or through an access device (which provides required interfaces). Subscriber access incorporates the implementation of a set of protocols (conventions) including mandatory use of the Internet Protocol (IP), the Transmission Control Protocol (TCP), and either the X.25 or ARPANET Host Interface Protocol, all of which are described later. [Ref. 1: p 4-1]

B. DEFENSE DATA NETWORK PROTOCOLS AND FUNCTIONS

1. DDN Protocols

The protocols in use within DDN establish general, minimum specifications (a convention) for any hardware/software combination to be connected to the DDN. By providing these conventions for each function of the network, the designers of the DDN have provided the latitude so that various common-user networks, using different hardwares/softwares, may connect to the backbone network with no modification to that network's upper level functions. The protocols are described in great depth in military standards and packet switching node specifications. [See references 4,5.] The fol-

<table>
<tr><th></th><th>OSI</th><th>DDN</th></tr>
<tr><td>7</td><td>Application</td><td rowspan="3">TELNET
FTP
SMTP</td></tr>
<tr><td>6</td><td>Presentation</td></tr>
<tr><td>5</td><td>Session</td></tr>
<tr><td>4</td><td>Transport</td><td>TCP</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Network</td><td>IP</td></tr>
<tr><td>X.25 AHIP</td></tr>
<tr><td>2</td><td>Data Link</td><td>Data Link</td></tr>
<tr><td>1</td><td>Physical</td><td>Physical</td></tr>
</table>

Figure 1. Comparison of Layering in the OSI and DDN

lowing descriptions will attempt to capture only the more salient features of the protocols of interest.

The protocols of interest for the model proposed in this thesis include the Internet Protocol (IP), the Transmission Control Protocol (TCP), and the network level protocols and functions. The network level functions are part of the backbone network of the DDN and include operation of the packet switching nodes (PSN) and interswitch trunks (IST).

a. The Internet Protocol (IP)

The essential element of the DDN is the backbone network of links and packet switching nodes which permits the interconnection of numerous common user networks. Each common-user network will have its own system peculiarities as a result of its internal upper level protocols. The only requirement on those protocols is that

they provide minimal datagram service; that is interface with the lower level functions provided by the DDN. As such, a methodology was required which allows two different networks to exchange data with the IP as the intermediary. The IP operates at the network level below the transmission control protocol, which is described later.

Upon receipt of a message from the TCP for transmission to a destination host, the local IP encapsulates the data in a datagram. If the message is too large to fit within a single datagram it may be broken into several datagrams, however, the basic unit of transfer is the datagram (packet). A datagram is a conveniently transportable piece of the total message being transmitted host-to-host. That is, a relatively long message is broken into segments which are an appropriate size for transport over the network. In order to successfully transport this segment the IP encapsulates the segment within a frame which holds sufficient information to reliably transport the segment from the sender to the destination without reference to any other segment of the message. The datagram must then allow for reassembly of the segments into the original message upon receipt by the destination host. This encapsulated segment is the datagram. The datagram will include header information such as destination, type of service, checksums, source, and length. If the destination is on the same network, the datagram is sent PSN-to-PSN to the destination, where it is passed to the destination host's upper layers. If the destination is on another network, the IP will identify a gateway which provides a transition point between the source network and the destination network (or a network which serves as a link between the source network and the final destination network). The datagram is routed over the network to the gateway, where the gateway IP extracts the IP data from within the initial network's datagram and encapsulates it in the subsequent network's local protocol datagram. The gateway then submits it to that network for transfer to the destination. The encapsulation within the subsequent network's datagram may include fragmentation of the data, if the second network's local datagram size is smaller then that of the first. The IP provides for this fragmentation and subsequent reassembly. [Ref. 6: pp 262-263]

This particular aspect of the IP (specifically datagram size) may be an area for application of operations research techniques; for example determination of the optimal datagram size in a network where large datagrams may result in more damaged datagrams in transit and small datagrams will result in excessive overhead.

The IP provides additional services which include *time to live, header checksum, type of service* and *options.* Time to live puts a limit on the lifetime of a datagram, so that datagrams do not remain in the network indefinitely. Header

checksum provides for reliable transfer of header information to ensure accurate transfer; it does not provide for data verification. In either of these services, failure results in the destruction of the datagram. Type of service refers to quality of service desired, such as precedence level, and interactive, real time or bulk service. All of these types of service may be dependent upon the capabilities of the common-user network itself. Options include *timestamps, error reports* or *special routing*. [Ref. 6: pp 263-264]

b. The Transmission Control Protocol (TCP)

The Transmission Control Protocol is designed to provide reliable datagram delivery in a data communication network environment in which loss, damage, duplication or misordered data can occur. The Transmission Control Protocol (TCP) used by the DDN is mandatory for use by all users of the DDN.

The TCP operates at the transport level, as the interface for upper level protocols, and above the Internet Protocol (IP). Because TCP operates above protocols which do not necessarily provide high reliability service, it must compensate for this potential unreliability. The mechanisms used by TCP to ensure reliable service include such mechanisms as error detection, positive acknowledgements, sequence numbers and flow control. [Ref. 4: p 1-168]

Positive acknowledgement with retransmission (PAR) is utilized to compensate for loss, or damage to datagrams in lower levels. This mechanism provides that unless a positive acknowledgement of receipt of a message (segment) is received by the sending host transport layer, it will automatically retransmit that segment upon expiration of a timeout interval. "In TCP, the timeout is expected to be dynamically adjusted to approximate the segment round trip time plus a factor for interval processing, otherwise performance degradation may occur." [Ref. 4: p 1-168]

Additionally, TCP utilizes simple checksums to detect damaged segments, which are then destroyed. This forces the PAR mechanism to compensate for such errors. This is a verification of the validity of the data itself, as opposed to that checksum found in the IP which only checks for validity of the header information. [Ref. 4: p 1-238]

The TCP utilizes sequence numbers to detect out-of-order and duplicate segments. By use of a sliding window scheme the TCP controls the flow of data from one host to another. This is accomplished by limiting the sequence numbers which are considered acceptable for transmission/receipt by the sender/receiver. As resources become limited at the receiver the size of the window is decreased, which forces the sender to slow the transfer of segments to that receiver. As acknowledgements are sent/received

by the destination/source the window is incremented to subsequent sequence numbers. If additional resources become available at the destination host, the window size is increased, allowing the source to increase the data transfer rate. As such, the integration of the positive acknowledgement and sequence number mechanisms provides for flow control. [Ref. 4: pp 1-169,1-234]

c. *Application Level Protocols*

Application level protocols operate at levels above the TCP and IP. Typical protocols at this level include file transfer protocol (FTP), TELNET protocol and simple mail transfer protocol (SMTP). The FTP is used to specify the format for transfer of data files between network hosts, and for allowing access to file manipulation functions. The TELNET provides for communication between terminals and remote hosts across the network. The SMTP provides for transfer of electronic mail over the network. Because these protocols operate above the levels used by the backbone network of the DDN they were not incorporated into the simulation model. [Ref. 6: pp 263-265]

2. The Backbone Network Functions and Protocols

The backbone network is made up of the packet switching nodes (PSN) and interswitch trunk (IST) lines. The backbone network uses unique protocols and functions; with the difference being that functions are carried out by the microcode of the PSN and thus are specific to that software and the protocols remain conventions to facilitate data communication between possibly dissimilar networks.

At each PSN several functions occur which provide interface between adjacent PSN's and with host-type devices. Some of these functions are PSN management functions and are only described briefly. The balance make up the network lower level functions of the DDN and are of greater interest to this model.

Management functions include PSN initialization, line up-down verification, statistics collection, monitoring, reporting and control, and PSN termination. PSN initialization and termination provide for the startup or shutdown of a PSN upon repair or detection of faults. Line up-down verification monitors the status of trunk lines to adjacent PSN's. Statistics collection collects data on throughput and utilization, which are then used by the monitoring, reporting and control function.

Other functions of greater interest include resource management (RM), host interface (HI), end-to-end (EE), congestion control (CC), store-and-forward (S&F), link control (LC) and routing. Host interface protocols of interest within the backbone network are X.25 and ARPANET host interface protocol (AHIP). Other protocols of in-

terest within the backbone network are the end-to-end protocol and the logical channel protocol.

"The X.25 protocol governs communications across the interface between an X.25 host and a PSN. It provides the network level and link level interfaces necessary for host and PSN communication." [Ref. 5: p 4-13] This protocol interfaces with the end-to-end function at each host interface PSN. It includes support of logical addresses in two modes: physical addressing (the use of specific physical node and ports to which connection has been made) and logical naming (transparent routing by use of names associated with one or more physical addresses). Additionally X.25 allows negotiation of common packet and window sizes between other X.25 hosts. It will support X.25 host to AHIP host connections. [Ref. 5: p 4-15]

A.. .?ANET host interface protocol (AHIP) supports communications between an AHIP host and its interfacing PSN. It includes two versions, local host (LH) and distant host (DH). It is implemented at OSI layer 3, and interfaces with the end-to-end function at the PSN. Implementation consists primarily of the proper acceptance and buffering of messages to upper levels and status (up/down) monitoring of the host. [Ref. 5: p 4-24]

The end-to-end protocol is analogous to the TCP and thus provides for reliable communications between source and destination PSN's in a network. Because of the nature of datagram service, with the possibility of each datagram following different routes, at different speeds and possibly arriving out-of-order, the EE protocol ensures that the source host's interfacing PSN relays error-free messages to the destination host in the same order as sent by the source host. The source PSN EE functions to packetize messages from the host, buffer and retransmit messages unless acknowledged by the destination's PSN, to transmit messages to the destination host sequentially, as well as to send internal acknowledgement (IACK) of the receipt of a message to the source PSN. The destination PSN sends an additional end-to-end acknowledgement (EACK) when the destination host acknowledges it has received a complete, verified message. [Ref. 5: p 4-26]

The logical channel (LC) protocol operates at the link level and functions as a means to obtain high reliability transfer of packets between adjacent PSN's. Reliability refers to resistance to noise on communication links; noise results in damaged packets. Efficiency and reliability may be in conflict as a result, with high reliability resulting in increased network overhead and reduced efficiency. [Ref. 5: p 4-28]

Efficient use of PSN-to-PSN links is accomplished by dividing bandwidth into full duplex logical channels (up to 128 logical channels per link). Reliability is improved by buffering each packet for retransmission at timeout intervals until the receiving PSN acknowledges receipt. Only one packet at a time is sent over a logical channel in order to minimize processing overhead. This allows simple toggling of transmit state tables to indicate acknowledgements and to identify duplicate packets. Acknowledgments are appended to regular packets directed to the appropriate PSN or, if no traffic is destined for that PSN, by generation of a null packet. [Ref. 5: pp 4-29,4-30]

a. Packet Switching Node Computer Program Functions

Functions are the capabilities provided within the PSN computer programs. As mentioned previously these functions can be broken into network management and network operations. The functions of interest are those concerned with network operations and include resource management, host interface, end-to-end, congestion control, store-and-forward, link control and routing.

"The resource management function (RM) controls the distribution and use of allocatable memory resources, buffers, and connection blocks." [Ref. 5: p 5-7] Within this function there are guaranteed minimum allocations for specific functions. To improve resource utilization, like functions are pooled and may share resource allocations. Finally, there are free resources for use by functions with high current demand. Per host input/output pools guarantee each host a secure input or output pool for end-to-end connections. The transit pool provides dedicated buffer resources for PSN-to-PSN traffic, and prevents loss of packets resulting from excessive source traffic. [Ref. 5: p 5-10]

This is another area which may be well served by operations research techniques particularly in the area of determining optimal allocation of buffer space to specific functions; and determining if there should be a priority assignment for buffer allocation during high load periods which may improve network performance.

The host interface function (HI) provides for the boundary operations necessary for successful communication between various host type devices and the backbone network. It provides for interoperability between various host interface protocols such as X.25 or AHIP. HI acts as the interpreter between the end-to-end protocol and the host-device protocol, while passing complete messages to the destination host, or receiving messages from a source for transmission on the backbone network. Messages from hosts served by a single interfacing PSN are processed on a first-come, first-served basis. [Ref. 5: p 5-35]

The end-to-end function (EE) provides for the management of message transmittal/receipt from source/destination PSN's. EE provides message assembly/disassembly, sequencing, checksum verification and positive acknowledgement and retransmission to ensure reliable transmission on a source-to-destination basis. [Ref. 5: p 5-48]

Upon receipt of a packet from store-and-forward function, any appended acknowledgements are stripped and processed. If the packet is the first to arrive in a multi-packet message EE requests a buffer allocation from RM for the storing of the entire message. Insufficient buffer space for the entire message causes the packet to be destroyed, except in cases when it is an older sequence numbered message from a host which has buffer space allocated to newer sequence numbered messages. In that case the newer sequence number's buffer would be reallocated to the older sequence numbered message causing destruction of packets from that newer sequence numbered message. If the packet has previously been received, it is destroyed. If the packet is of a message that is out of the current sequence number range, it is destroyed. [Ref. 5: p 5-56]

The implementation of this reservation scheme at the destination may no longer be in use within the DDN. Recent conversations with Defense Communication Agency representatives raised the question of whether this function may have been recently deleted. However, this could not be documented.

The packet making the buffer reservation becomes the basis for message reassembly. Upon receipt of all packets of a message, checksum verification is performed. If an error is indicated, the message is destroyed. Otherwise, EE submits the message to HI (in sequence if required). Upon transfer to HI, EE sets up an internal acknowledgement (IACK) for transmittal to the source PSN. An external acknowledgement (EACK) is transmitted when the destination host acknowledges receipt of the message. Receipt of an EACK increments the flow control window.

Upon receipt of a message for transmission, EE packetizes the message (if required), submits the packet(s) to congestion control, and activates a retransmission timer. [Ref. 5: p 5-58]

The congestion control (CC) function provides for efficient utilization of backbone network resources, in particular those associated with PSN-to-PSN traffic. To accomplish this, it receives and stores information sent by the network PSN's regarding their S&F utilization, transmits information regarding its own S&F utilization, rations packets sent along paths for which data indicate high utilization of S&F re-

sources and allocates its own S&F resources in response to network demand. CC utilizes the information received from other PSN's to determine the amount of traffic which can be supported by the routes to each destination and will throttle the traffic submitted by EE based upon that determination. [Ref. 5: p 5-68]

The store-and-forward (S&F) function controls the flow of all traffic through the PSN, that is PSN-to-PSN traffic from link control (LC) or host traffic destined to, or received from, EE. Each of these types of traffic has its own queue in front of S&F processing. S&F processing alternates between the queues to ensure equitable throughput. [Ref. 5 : p 5-73]

Upon receipt of a PSN-to-PSN packet from LC, S&F will determine the logical channel it arrived on, then determine if the packet is a duplicate. If it is a duplicate, S&F destroys the packet and retransmits the node-to-node acknowledgement. If the packet is not a duplicate, S&F submits it to LC for transmission to the next PSN on its route. Routing is determined by a call to the ROUTING function. S&F submits the packet to EE if the packet destination PSN is the current PSN. [Ref. 5: p 5-76]

If the traffic is from host input (EE), S&F performs dispatch tests which include: availability of logical channels, destination dead, and destination PSN equals current PSN tests. S&F submits the packet to LC for transmission to the next PSN on the route if all the tests are met. [Ref. 5: p 5-80]

The link control (LC) function performs packet receipt and transmission for PSN-to-PSN traffic. Priority processing is given to incoming traffic to minimize packet losses. Incoming traffic is then error checked and routed to appropriate follow-on functions such as S&F. If the incoming traffic is a node-to-node acknowledgement, LC releases that logical channel for reuse and frees the buffer space allocated for the retransmission process. [Ref. 5: p 5-84]

Packets received from S&F for transmission to adjacent nodes are assigned to a logical channel for transmission and buffered for retransmission upon completion of a timeout interval, unless previously acknowledged.

The routing function (ROUTING) develops and maintains the table consulted by the S&F function for packet routing information. The table indicates whether the destination can be reached, and the next PSN in a shortest-path algorithm. The routing table is updated using delay information transmitted via congestion/routing update packets (CRUP) or through delay measurement performed by the ROUTING function itself. Delay consists of intra-PSN delay (processing delay), transmission delay and propagation delay. Actual delays over each line are measured, by use of time

stamps in packet headers, and averaged every ten seconds. If the current average delay differs from the previous delay, plus or minus a threshold value, a CRUP is generated and flooded over the network. Upon receipt of a CRUP the ROUTING function updates a data structure which holds the minimum delay paths to all PSN's in the network. The routing table is derived from this data structure. [Ref. 5: pp 5-113,5-115,5-118]

III. THE MODELING LANGUAGE-SLAM II

A. GENERAL DESCRIPTION

The modeling language chosen for use in the development of this simulation was SLAM II (Simulation Language for Alternative Modeling). SLAM II is an advanced FORTRAN based simulation language which allows modelers to approach problem solving from a process, event or continuous modeling orientation. The language has been given sufficient flexibility to handle any combination of discrete (process, and/or event) and continuous models into a single model. [Ref. 7: p ii]

The process orientation portion of SLAM II is a network structure which uses predefined statements to model flow through a process. The language refers to these predefined statements as nodes and branches. Examples of nodes include QUEUE, ASSIGN, CREATE, and ACCUMULATE nodes. These nodes establish the particular process element to be executed by the use of a simple command. For example: QUEUE will implement the code necessary to establish a waiting line or queue including file definition, statistical collection, and entity linkage for processing priority (i.e., FIFO, LIFO, etc.). Branches provide for flow between nodes and may be based on a decision, probabilistic branching or flooding of entities, and may include process delay. The network structure employs graphic representation of nodes and branches as an aid to efficient, correct modeling.

The event orientation is a discrete event structure which employs SLAM II subroutines for common simulation tasks such as calender operations, statistics collection and random sample generation. Events are defined by the modeler in user-written FORTRAN subroutines. The executive control program of SLAM II controls the occurrence of events and the simulation clock.

The continuous orientation of SLAM II utilizes user-written differential equations which describe the dynamics of the model. SLAM II provides for specially defined storage arrays and executive control of the simulation.

B. ADVANTAGES AND DISADVANTAGES OF THE USE OF SLAM II

One of the distinct advantages of SLAM II is that any of the process, event or continuous orientations may be used in combination with the others. For instance, in the model under study, the network structure of SLAM II is able to provide most of the underlying framework for the model. In places where the network does not provide the

specific processing desired, an EVENT node is utilized. This provides a transition to the event orientation structure, where a FORTRAN subroutine is executed to provide the desired processing. Upon completion of the subroutine SLAM II returns to the network structure at the event node. In this manner, entity attributes or model global variables may be processed, at appropriate points in the network, without disrupting the flow of the network.

SLAM II was chosen as the modeling language for a number of reasons. The primary reasons are these: it is a FORTRAN based language, and the modeler was already adept at programming in FORTRAN; it is a language which provides high level simulation functions yet allows access to the underlying lower level functions and code so that the modeler is able to write those functions not provided by SLAM II or to add detail to functions at a greater depth then SLAM II provides; the network structure provides the majority of process elements needed for the model under study; SLAM II was immediately available for use on the Naval Postgraduate School (NPS) mainframe.

As the modeler became proficient in SLAM II, several other advantages became obvious, including: SLAM II is a language that is easy to learn; the graphical representations aid in conceptualizing the model, its flow, and the appropriate coding of the model; debugging messages are clear and to the point; and the model may be made portable by purchase of the microcomputer version of SLAM II.

SLAM II has these disadvantages currently: very limited local NPS experience with SLAM II on the NPS mainframe which caused problems to be discussed later; there is very little consulting assistance available for this language at NPS; execution speed is very slow; and the statistical package included with SLAM II does not provide as much depth as was desired.

C. IMPLEMENTATION PROBLEMS

During the modeling and analysis process a few problems were encountered with the use of SLAM II, as currently installed on the NPS mainframe. First, the dimension of data arrays is extremely limited and is really only sufficient for small models. Changes in the simulation model code resolved this problem for this model, and a general resolution, for all users, was accomplished by requesting the computer center to recompile the SLAM II code. This resolution was the result of an NPS conversion from a Version 1 FORTRAN compiler to a Version 2 FORTRAN compiler. During the transition phase to the new compiler the author noted that Version 1 compiled code would not execute under Version 2, as it should have, when used in conjunction with SLAM II code, as

installed on the mainframe. This forced the computer center to recompile the code under Version 2. At the same time the computer center expanded the dimensions of the previously discussed data arrays. Prior to the recompilation under Version 2 it was brought to the computer center's attention that the execution time was extremely slow for large models. At the prompting of the author, it was discovered that SLAM II was originally installed/compiled using FORTRAN compiler optimization level 0. When SLAM II was recompiled under Version 2, it was compiled at optimization level 3. A very cursory investigation into the improvement in execution speed provided by this higher optimization level showed that there was approximately a 33 per cent improvement.

SLAM II is an excellent language for use within the Operations Research Department at the Naval Postgraduate School and should be immediately incorporated into the curriculum. The advantages are clear and most of the disadvantages were resolved in the thesis process or are overcome with simple FORTRAN subroutines.

IV. THE SIMULATION MODEL

A. GENERAL DESCRIPTION

A stochastic simulation model has been written in the SLAM II simulation language to capture pertinent aspects of the DDN protocols in order to study network dynamics and performance. Because of computer resource limitations it must have a drastically reduced topology, as compared to the DDN, which consists of hundreds of packet switching nodes dispersed essentially world-wide. The SLAM II code and supporting FORTRAN subroutines are listed in Appendix A.

The network studied consists of four hosts communicating over a backbone network consisting of five packet switching nodes (See Figure 2 on page 17). This topology was chosen to allow both cross and reverse flow traffic in "conflict" with the traffic of interest. In addition, it allows for more than one path of transmission between hosts.

After a message is accepted into the system it, (all of its packets), must transition from origin to destination. The time required to do so is called the *response time*. The object of this thesis is to study the effect of retransmission protocols on the response times.

A typical message arrives at its host and is divided into packets. The packets then await space in a sliding window scheme and, after proceeding through the window, enter the subnet. Simultaneously, a copy of each packet enters a retransmission loop which will cause retransmission of the entire message upon expiration of a host-to-host timeout clock unless an end-to-end acknowledgement is received first.

Upon entering the subnet each packet enters a queue (with dedicated space for this host) at the interface PSN. The queue is served by a single processor which transmits the packet to the next PSN, as chosen by a routing algorithm. The packet simultaneously enters a second retransmission loop at the PSN. This will cause retransmission of the packet upon expiration of a node-to-node timeout interval unless a PSN-to-PSN acknowledgement (ACK) is received first. If space is available in that subsequent PSN buffer, the packet enters that queue, otherwise it is destroyed. This process continues PSN-to-PSN until the packet reaches the PSN which is the interface for the destination host, where the packet is sent to the receiving side of that host.

Upon entering the receiving host, a test is made to determine if this packet is part of a message eligible to be received, that is, is within the sliding window. If it is, then a

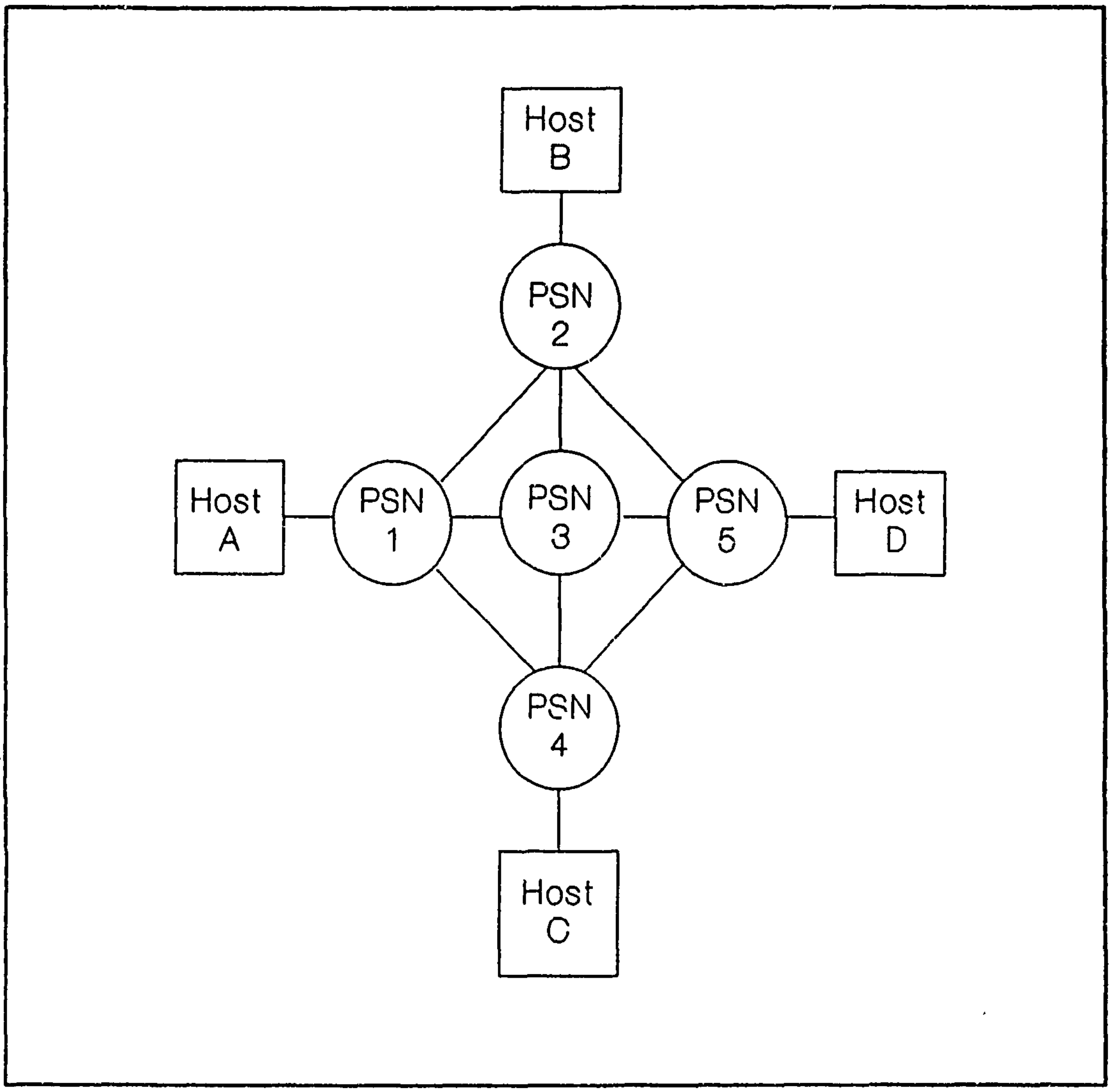

Figure 2. Topology of the Modeled Data Communication Network

test is made to determine if buffer space has been reserved for the entire message. If not, the packet attempts to make a reservation. If buffer space is not available the packet is destroyed. If a reservation has previously been made, or after one is made, the packet awaits arrival of all other packets belonging to the same message, at which time an end-to-end acknowledgement (EACK) is sent, the sliding window is incremented and the message relayed to the upper layers of the host.

The simulation attempts to capture several significant features of the DDN including:

1. Retransmission of messages, host-to-host, unless acknowledged;
2. Retransmission of packets, PSN-to-PSN, unless acknowledged;
3. Limited buffer space at receiving hosts for incoming messages;
4. Limited links between nodes;
5. Reservation of receiving host buffer space by the first arriving packet of a message;
6. Processing of packets from any one message is delayed until all packets of that message have arrived at the receiving host.

B. DETAILED DESCRIPTION

1. Message Arrivals

(See Figure 3 on page 19)

Message arrivals from outside the system to a sending host are modeled using the SLAM CREATE node. Of significance here is that the CREATE node schedules the next arrival by use of an interarrival delay. To be specific, independent random draws from an exponential distribution determine the interarrival times of members of the sequence of message arrivals. This will tend to make traffic "bursty", that is, a lot of periods of short interarrival times between messages, with occasional long interarrival times between messages. It was felt that this would most truthfully model actual message arrivals, where a user might hold a brief "conversation" (burst of messages) with the receiver with a relatively long time before the next "conversation". [Ref. 8: p 35]

The simulation model assigns attributes to each arriving message by use of SLAM ASSIGN nodes. These attributes hold the characteristics of each message/packet in the network for use in statistical analysis and network decisions. The attributes include: time of arrival (ARRIVAL), source host (SENDER), destination host (RECEIVER), current node location (NODE), message serial number (MESSAGE), number of packets per message (PKTS), packet number (PACKET), and number of retransmissions (RETX).

Each arriving message's destination is determined by use of SLAM II probabilistic branching; such that, in the long run, a user initialized fraction, p_{ij}, (where i=A,B,C,D; j=A,B,C,D; and $i \neq j$) of the total traffic from source i is sent to destination j. The p_{ij} are initialized at the start of each simulation run. In this thesis p_{AD} is generally held at 1.0. The reason for doing this is that a significant amount of the traffic should be made to conflict crosswise and opposite to the flow of the traffic of interest (A to D) in order to provide realistic network dynamics. In other words, there must be

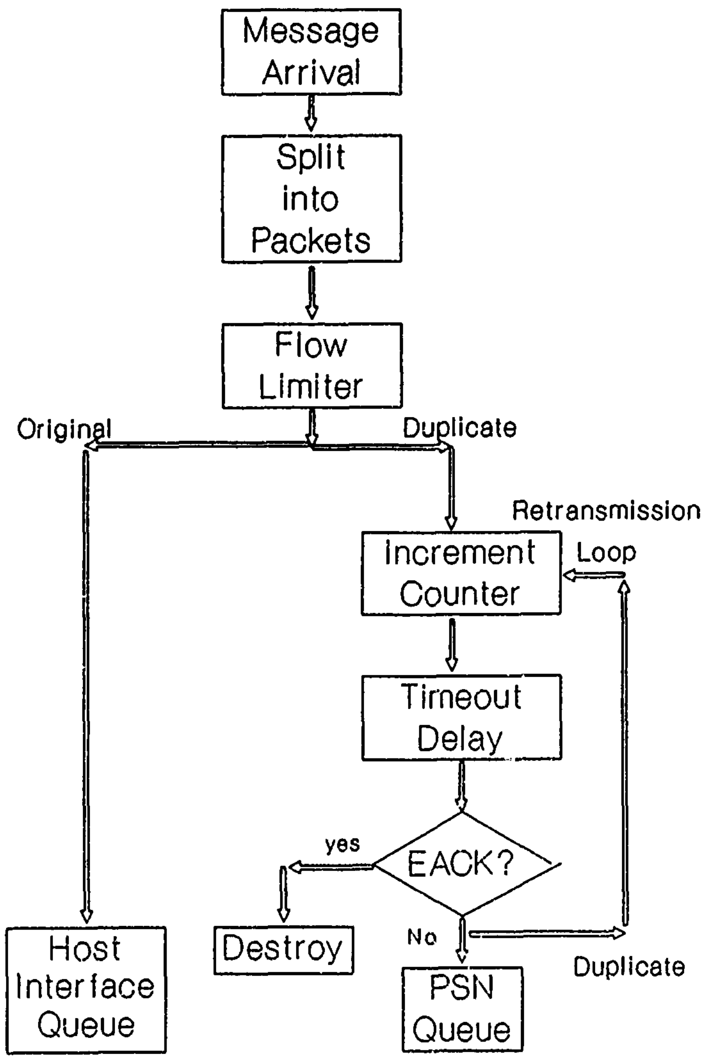

Figure 3. Flow Diagram of a Typical Message Transmission

competition for subnet assets (nodes, links, queues) as well as competition for the buffer space at each host's receiving side.

The message is then divided into packets and proceeds to the sliding window. The simulation requires that the number of packets into which each message is divided be fixed, and determined prior to running the simulation. This requirement allows files and queues to be established within the simulation code. The number of packets per message is set at three, although this number could be changed by altering the simulation code. The sliding window is modeled by use of a SLAM AWAIT node and SLAM files which hold the message serial numbers within each window. The AWAIT node is essentially a resource limit, with each packet utilizing one resource. This causes each host to have a limit to the number of messages it has on the subnet at any one time. By keeping two files of the message serial numbers which have proceeded through the AWAIT node, the source and destination can each independently determine whether a message has been acknowledged or previously received in entirety. This arrangement mimics the DDN sliding window, for hosts and receivers, and, through the use of appropriate delays (discussed later), simulates end-to-end acknowledgements.

Each packet is then duplicated, with the original proceeding to a queue at the interface PSN, which has buffer space dedicated to this host. If this buffer is full, packets are held at the host until space becomes available. Any such waiting packets are immediately placed into the host interface buffer as space becomes available.

The duplicate enters a retransmission loop. Within this loop the RETX attribute is incremented to count the number of retransmissions required to successfully complete transmission of the message to the receiver. The duplicate packet then is delayed for a period of time equal to the host-to-host retransmission timeout. Upon completion of this delay event 4 is called. This is a test to check if this message is still in the sender's sliding window file; that is, whether an acknowledgment has been received. If an acknowledgement has been received then the duplicate packet is destroyed. Otherwise it is again duplicated, with one copy reentering the retransmission loop and one copy being sent to the subnet queue side of the interface PSN. This is significant because if the subnet queue is full then packets are lost, whereas the originals are stored if their queue is full.

It should also be noted that because of the delay in the end-to-end acknowledgement transiting back to the sender, it is possible for retransmission of a message to occur <u>after</u> a message has been received in entirety by the destination host.

2. Packet Switching Nodes

(See Figure 4 on page 22)

Upon arrival at a PSN a packet attempts to enter either the subnet queue or the host queue (as appropriate), modeled by two SLAM QUEUE nodes. As previously mentioned the host queue occupies buffer space dedicated to host-to-subnet interface traffic. The subnet queue is used by any PSN-to-PSN (subnet) traffic or a host retransmission. Two queues were modeled because the DDN dedicates buffer space to particular uses [Ref. 5: p 5-10]. In addition it allows the model to 1) block traffic when the host queue is full; that is to force the host to hold backed up traffic without loss, and 2) to balk traffic if the subnet queue is full; that is to cause packets to be destroyed if the subnet queue is full. Each queue is currently assumed to have space for ten packets. This number is somewhat arbitrarily chosen so that under moderate traffic loads the queues will occasionally completely fill, causing loss of some subnet traffic. Of course it is easily changed.

Processing at the PSN is modeled by a single server which alternates service between the two queues. The server is modeled by use of the SLAM SELECT and service ACTIVITY nodes. Processing time can be initialized at the start of each simulation and is currently set to 0.012 seconds, which mimics a 675 bit packet being sent out on a 56 KBPS transmission line [Ref. 9: p 4]. Upon completion of the processing delay the packet attempts to occupy a PSN-to-PSN transmission link modeled by use of a SLAM AWAIT node.

The AWAIT node limits the number of packets in transmission from this PSN to any other adjacent PSN (limited to 16 times the number of adjacent PSN's i.e., 48 for PSN's 1, 2, 4, and 5 and 64 for PSN 3). This simulates the logical channels on each link between DDN PSN's, which, in the actual DDN topology, may be up to 128 channels per link [Ref. 5: p 4-28]. No attempt was made to model an AWAIT node for each PSN-to-PSN arc because, owing to the routing algorithm characteristics, a PSN will see virtually equal traffic outbound on each arc and thus, use of the (say 48) links would be generally spread evenly between the (3) outbound arcs.

The packet then proceeds through some recordkeeping at SLAM ASSIGN nodes, at which the current PSN location and previous PSN location are updated.

The packet next calls the routing algorithm (described in detail later) where the next PSN along the route is determined based upon the destination of the packet and the shortest wait in queues along possible routes. This event also puts this packet in a

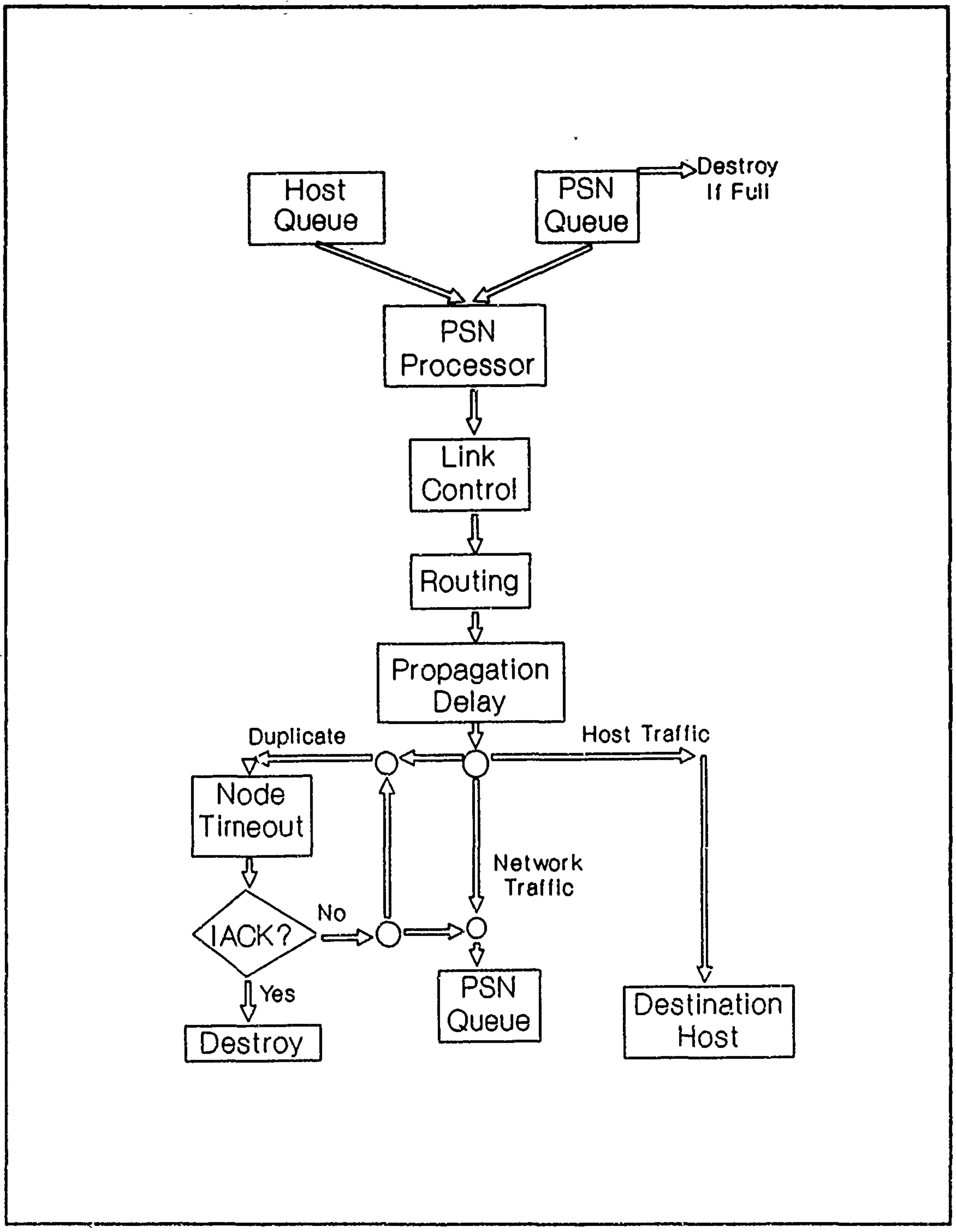

Figure 4. Packet Switching Node Flow Diagram

file that contains a list of those packets which have been transmitted from this PSN but not acknowledged (PSN-to-PSN).

The packet is then duplicated, with the copy being sent to the PSN-to-PSN retransmission loop, where it delays for the node retransmission timeout. Upon completion of the timeout delay, event 9 is called which tests for presence in the file which holds the list of unacknowledged packets. If the packet is not listed it is destroyed. If it is listed it is retransmitted to the same PSN as previously chosen by the routing algorithm (even if the path routing has been updated in the meantime) and a copy reenters the retransmission loop.

The original copy of the packet proceeds to a SLAM II activity node where it is delayed for a time period which simulates both a propagation delay and a processing delay at the next PSN. This time period is initialized at the start of the simulation and was set at 0.001 seconds for this study. By use of SLAM ACTIVITY nodes a one percent error rate is introduced between PSN's. This error rate simulates transmission errors caused by electromagnetic bursts on the transmission lines. Upon completion of this delay event 8 is called. This event checks to see if there is room for the packet in the next PSN's subnet queue (if the packet is staying on the subnet). If there is space, then the current link is freed (releases a link resource at the previous PSN's AWAIT node). In either case the packet is sent to the next PSN where it enters the subnet queue or is lost if the subnet queue is full. If the next point in the route is the destination host the packet is sent to the receiving side of the host vice any PSN queue.

3. Routing Algorithm

(See Figure 5 on page 24)

The routing algorithm used employs several SLAM functions and SLAM's ability to manipulate data in FORTRAN subroutines. At each PSN encountered in the transit from source-to-destination the routing algorithm is called, and the next PSN in the route is determined by entering the table with the current PSN location (i) and the packet's final destination host (j) (where i = 1,2,3,4,5 and j = A,B,C,D). Element a_{ij} of the table defines the next PSN that this packet should be sent to on its transit to the destination; otherwise it will indicate that the packet should be directed to the receiving side of the destination host.

Certain routings were actually fixed by the nature of the topology. First, if the current PSN is the interface PSN for the destination, the routing algorithm returns a zero indicating the route is to the receiving side of the destination host. Second, if the current PSN is PSN 3 (the center PSN) routing is fixed to go to the interface PSN of the

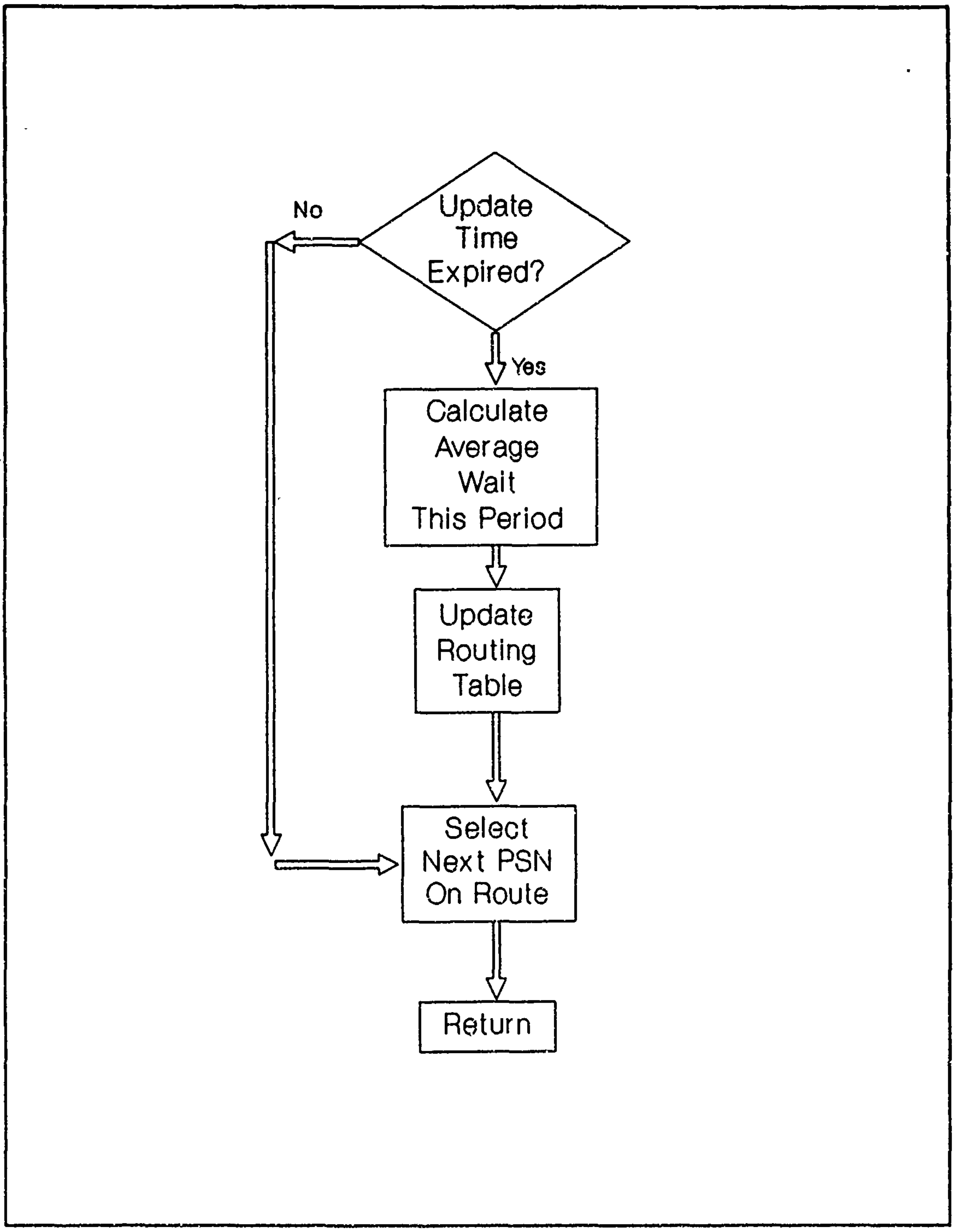

Figure 5. Routing Algorithm Flow Diagram

destination. This was done to prevent infinite loop cycling of the routing. Thirdly, if the packet is at a PSN (say PSN 1) and destined for its destination host via a diagonal arc (say host B) then packets are routed directly to that interface PSN (PSN 2). So the only updates occur at elements $a_{1,D}$, $a_{2,C}$, $a_{4,B}$, and $a_{5,A}$ of the routing table.

For the first ten second period of a simulation run, the next PSN in the route each packet takes from its present location to the destination host is arbitrarily fixed by initialization of a routing table. During this initial period elements $a_{1,D}$, $a_{2,C}a_{3,B}$, and $a_{5,A}$ are set to 2, 5, 1, and 4 respectively. Subsequently, the routing table is updated at approximately ten second intervals. The table is updated by computing the number of packets which had to wait in each PSN's queue and the total time spent waiting in each PSN's queue, in the period since the last update (approximately ten seconds).

The number of packets passing through the queue is computed by using the SLAM functions FFAVG and FFAWT. FFAVG returns the time weighted average number of entities in a file (queue) and FFAWT returns the average waiting time in a file (queue). From these and TNOW (current simulation time), N, the number of packets passing through the queue and WAIT, the total wait in that queue can be computed. By maintaining global variables which are set to the calculated values of N and WAIT the number waiting and their total wait in the last ten second period may be calculated iteratively. The average wait over the last ten seconds in each queue is then determined, and the table updated such that all traffic is routed to the shortest average wait queue. A ten second interval was chosen because it is the same time period that the DDN uses in normal updates of its routing tables [Ref. 9: p 2]. However this interval length can be adjusted if desired.

An additional assumption is that the routing table is instantaneously available at all PSN's upon update. This is not true for the DDN, where there is a time delay for the routing update packets to circulate through the entire network and thus update the routing tables at each PSN. However, the DDN topology is very large, accounting for the time delay for the full network to be updated, whereas adjacent PSN's receive CRUP packets virtually instantaneously from their immediate neighbors as a result of the priority handling of CRUP packets. Thus the small topology of this model would, at least initially, seem to justify the instantaneously available update. This routing algorithm is not as realistic as DDN's, where "whip lashing" was noted when a similar algorithm was in use in early development of the system, and where there is a bias built into the shortest path first algorithm in order to take into account the longer propagation times on satellite links. [Ref. 9: pp 4-5] It is felt that the topology and loading of this model

does not justify a more sophisticated algorithm, and that, at least initially, essential detail is captured despite the simplicity. This assumption may not hold true as the model is expanded to include additional PSN's or at extremely heavy traffic loads.

4. Receiving Host

(See Figure 6 on page 27.)

Upon being routed to the receiving side of the destination host the packet is first tested to determine if it is a packet of a message which has previously been received in entirety. This is accomplished by checking for the message serial number in the file which was used at the sliding window of the sending host. If the message transfer has already been successfully completed the packet is destroyed. Otherwise a test is conducted as to whether buffer space has been reserved for this message. The first arriving packet of any message attempts to reserve buffer space for the entire message. If buffer space is not available for the entire message the packet is destroyed. If buffer space is available for the entire message that space is reserved. Buffer size was arbitrarily set to allow up to ten messages to make reservations at hosts during the node timeout interval experiment. During the host timeout interval experiment host D has a maximum of two messages allowed into its buffer. This smaller value was chosen to force the buffer to completely fill under traffic loads which do not cause gridlock on the subnet. It could be assumed that the buffer space for another eight messages is occupied by traffic from hosts not explicitly modeled. Dependence of mean response time to buffer capacity is not explicitly studied in this thesis, although the simulation model developed here can be modified slightly to do so.

Once buffer space has been reserved (or if it previously had been reserved) then the packet is placed in a file based upon its packet number (sorted by packet number). A test is first conducted to allow only one copy of that packet number/message serial number combination in the file (duplicate copies are destroyed).

A SLAM MATCH mode is utilized to hold packets in their respective files until one packet with the same message serial number is in each file. When this occurs that (now complete) message is allowed to flow further through the network.

Once all packets of a message are received, the buffer space is released, the receiving host's copy of the sliding window files is updated to indicate that this message has been received in its entirety, and routing time statistics are collected. The message then pauses for a time interval equal to the routing time of the last message sent from the receiver to the sender. This simulates the time delay for an end-to-end acknowledgement of the message just received to get back to the originator. Upon completion

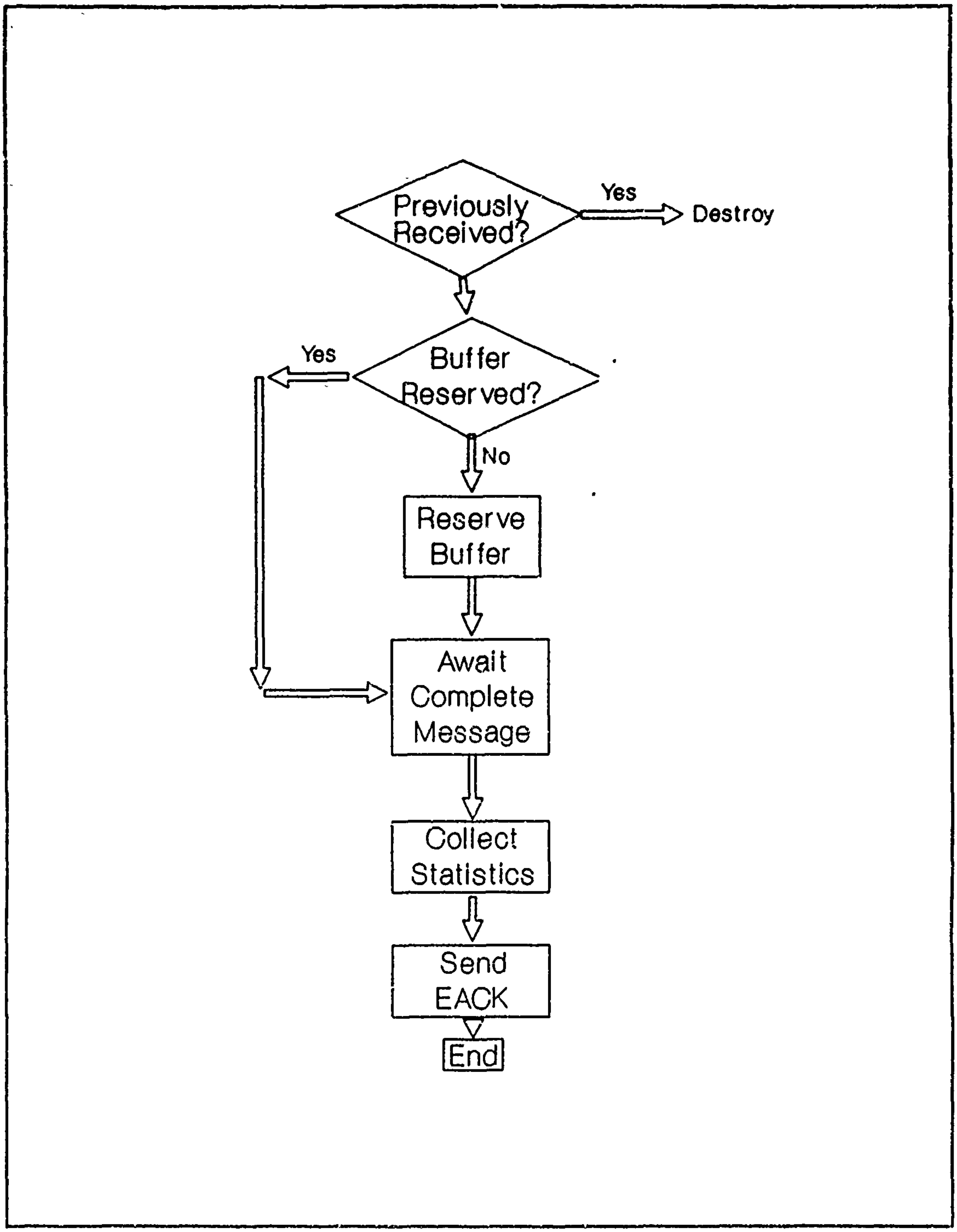

Figure 6. Flow Diagram of a Receiving Host

of this delay the sender's sliding window files and resource limiter are updated and the message is destroyed.

5. Simulation Techniques in Use

In order to reduce the variance of estimates from simulation data, the random number streams for message arrivals at hosts, and for error generation between PSN's, are common to all experiments involving the study of the effect of changing system parameters upon the routing time of messages from host A to host D.

V. ANALYSIS AND OBSERVATIONS

The analysis of the simulation model was attacked in essentially three phases, in a manner similar to that proposed by Law and Kelton [Ref. 10: pp 333]. The three phases being verification, validation and output analysis.

A. VERIFICATION OF THE SIMULATION MODEL

Verification has been defined as "determining whether a simulation model performs as intended i.e., debugging the computer program." [Ref. 10: pp 333-334] SLAM II provides for one of the most powerful techniques for use in this area: the *trace*. Other techniques include writing and testing the simulation program in modules, the use of structured walk-throughs and, finally, running the simulation model under simplifying assumptions for which performance, or steady state characteristics, can be calculated and used for comparison purposes.

The primary method of verification utilized was the trace technique. SLAM II utilizes a simple command which provides for a printout of the attributes of an entity, global variables and/or state variables each time an entity passes through the SLAM II modeling nodes. The printout can be for the entire simulation run or for any selected time interval of interest. The printout provides the node/branch label (if assigned), the node/branch type, and the desired attributes or variables [Ref. 10: p 281]. In this manner the actual flow of an entity through the model is traced, thus providing definitive evidence that the flow of the network is as the modeler intended and that value changes are being made at the appropriate time and to the appropriate value, as the modeler intended. Additionally, the contents of files can be printed at selected time intervals to determine that the contents of files are as the modeler intended.

As the size of the model grows, the trace report of full simulation runs becomes extremely cumbersome. This restricts the use of full trace reports to the initial phases of model development. In this thesis the trace report was used until the underlying framework of the model was completed. Thereafter, abbreviated trace reports were utilized to ensure that flow, through the SLAM II modeling nodes affected or implemented within embellishments, was as intended. The trace report was a valuable tool in debugging execution time errors and is considered a major advantage in the use of SLAM II as a simulation language.

An additional verification technique utilized was a modular program approach to the writing of the simulation program. Each section of code, for every host or packet switching node (PSN), is exactly the same, with the exception of node labels, storage files used and global variables assigned. This approach greatly simplified the code writing and debugging. Once a problem was identified and resolved in any one module, a duplicate correction in all like modules prevented similar errors in those modules.

In cases where SLAM II did not provide for desired network operations, FORTRAN subroutines were implemented by use of SLAM EVENT nodes. In each of these event subroutines, maximum utilization of SLAM II functions, variables and subroutines was employed, thus minimizing possible writer-induced errors. Additionally, each subroutine was kept as simple as possible. In those subroutines for which testing of results would be trivial (such as those for detecting presence of an entity in a file) no further testing was conducted. Such subroutines were further verified by use of a structured walk-through to be discussed later. In the more complex subroutines, such as the routing subroutine, simple yet comprehensive testing was conducted to ensure that the output of the subroutine was as desired.

The structured walk-through is the explanation of the code, by its writer, to a group of people familiar with the language and model, so as to convince them that the code does what the writer intended, or else to debug the code. There was limited use of the structured walk-through because there are few personnel at NPS who are familiar with the SLAM II language. It was, however, used several times in the debugging of simulation code. Professor Keebom Kang's assistance in this area was instrumental in solving several difficult debugging problems.

B. VALIDATION OF THE SIMULATION MODEL

Validation of a simulation model has been defined as "determining whether a simulation model is an accurate representation of the real working system under study" [Ref. 10: p334]. Validation has been considered a more philosophical rather then definitive procedure. Because this simulation model is not a representation of a specific real world system, but rather a simplified representation used to develop a better understanding of performance characteristics of a system, there is no "real world" with which to compare results. However, this does not prevent the analyst from examining results to determine if they make intuitive sense. For example, in early examination of the mean time to complete transmission of a message from host A to host D, under various host-to-host retransmission timeout intervals, it was noted that with very short retransmission time-

outs, the mean response time rapidly increased until the network went into gridlock (described later), at which point no messages got through. As the timeout interval was increased the mean response time gradually decreased to a fairly constant value. The initial decrease in mean response time that was noted as the timeout interval increased made sense because of the corresponding reduction in network congestion. However, it was felt that at some point the mean response time should again increase, due to the increased delay in completing transmission of messages which legitimately required retransmission. Closer examination of the SLAM II output seemed to indicate that while packets were lost between PSN's (because of transmission medium errors or full buffers) they would be retransmitted as a result of the logical channel protocol. In addition it was observed that buffer space at the receiving host was so ample that packets were never lost because of the inability to reserve buffer space for all the packets of a message. It was not until the end-to-end buffer space was reduced and packets were unable to make buffer space reservations that the anticipated upturn in the mean time to complete a message was noted.

Further validation was accomplished by generating output from the routing subroutine which gave the decisions made by the routing subroutine. The output was then compared to the values indicated by SLAM II functions which gave the average wait in each queue and provided a comparison value to determine whether the subroutine had made the correct decision.

C. ANALYSIS OF OUTPUT

In this study interest concentrates on the effect of timeout intervals on the mean time to complete a message form host A to host D, *the mean response time.* Because this performance measurement is best described as a steady state measure, *batching* (or sectioning) of the observed data was felt to be an appropriate methodology for estimating that mean response time. Proper use of sectioning gives the advantage of obtaining multiple, effectively independent observations of the response time from which to form an estimate and confidence intervals, from a single simulation run. Additionally, there is reduced influence from any transient phase caused by the initial conditions of the simulation run.

The major concern in the use of sectioning is that the size of each section or batch (m) is sufficiently large so that the correlation between any two observations, m observations apart, is nearly zero (assuming a covariance stationary process); in this study an observation is the time to complete transmission of a message from origin to destination

(in particular from host A to host D) but not including the time for an EACK to transit to the message originator. The literature gives a variety of methods for estimating an appropriate batch size (m) and the number of batches (n) to ensure a reasonably precise estimate of the mean response time and to assess the variance of that estimate.

Lewis and Orav present a rough rule of thumb which suggests that n be between 12 and 20 batches and if the total number of observations (N) is greater then 1000 then a batch size of m=N/n "will be big enough to make bias small and satisfy normality assumptions...." [Ref. 10: p 262]

Welch suggests the use of the autocorrelation function (ACF) to estimate the point at which the correlation between any two observations, k observations (lags) apart, approaches zero. Let L be some integer such that $\rho(k)\approx 0$ for $|k| > L$. A batch size (m) should be selected such that m >>L allowing an assumption of zero correlation between the i^{th} observations in any two adjacent batches. This will allow an assumption of effectively independent estimates of mean response time from each batch. [Ref. 10: pp 305-306]

IBM's Grafstat data analysis package implements the ACF and allows the plotting of confidence intervals for the estimated correlations at various lags. One such implementation is shown in Figure 7 on page 33 for a simulation run with 1000 observations of the time to complete a message from host A to host D. The observations are taken with simulation parameters set as in the node-to-node experiment (described later) and starting at simulation time zero under the assumption that there is no apparent transient phase for this model. This assumption is justified later. The estimated correlations ($\acute{\rho}(k)$) are plotted for lags k=1 to 100 along with the 95% confidence intervals for $\rho(k) = 0$.

From this figure it can be clearly seen that the correlation can assumed to be zero well before 100 lags. Consequently, batch size was, conservatively, set at 100 observations. During data collection, runs of 500 simulation seconds produced 15-20 batches, which corresponds with the Lewis and Orav rule of thumb discussed earlier. Additional plots of the ACF and associated confidence intervals are shown in Appendix B.

The next area of analysis is the determination of the approximate end of the transient phase caused by the initial conditions. Beyond this point the simulation is assumed to be in steady state. By neglecting any data from the transient phase we can better es timate the mean response time. The problem is selection of the time point at which to start collecting data such that the initial conditions are sufficiently uninfluential.

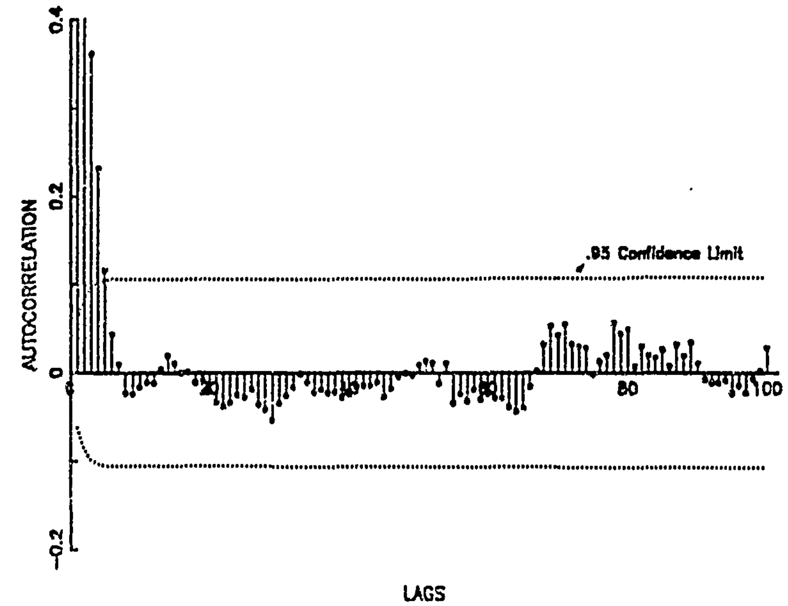

Figure 7. Autocorrelation Function Display for Use in Determining Batch Size

Welch suggests making a series of M pilot runs, each with N observations and graphing the sequence $\overline{V}_n$ (where: $\overline{V}_n = \frac{\sum_{m=1}^{M} V_{mn}}{M}$ and V_{mn} is the n^{th} observation in the m^{th} run, n= 1,2,...,N , m= 1,2....,M). As simulation time progresses the $\overline{V}_n$ will tend to approach some value in the limit (i.e., approach steady state). A plot of the $\overline{V}_n$ versus time (or observation number) should give an indication of the point at which the observation is sufficiently close to that limit to permit the assumption that the transient phase is over. Welch further suggests that fitting a smoothed curve to these data provides an aid

in selection of a truncation point beyond which the simulation may be assumed to be in steady state. [Ref. 11: p 282]

The result of this procedure is shown in Figure 8 on page 35. The data displayed includes M = 23 independent runs during which the first 1000 observations (N) of the time to complete a message from host A to host D were collected. A fitted line using the Lowess Function of Grafstat is also plotted, with the closest two per cent of the observations applied to the smoothing function. It can be seen from this figure that there is no apparent transient phase for this simulation model. Consequently data collection begins at simulation time zero (which includes the period of time during which the routing table was at its initialized values).

At first this may seem counterintuitive, in that it would be reasonable to expect to have some transient phase while buffers and queues filled. However, the nature of the topology and routing algorithm causes the steady state condition to be the condition during which buffers and queues are repeatedly and quickly filling and emptying, and thus the assumption of a very short transient phase is reasonable.

D. OBSERVATIONS OF MODEL DYNAMICS

A great deal of the effort in evaluating this simulation model was in garnering insight into the dynamics of the network in operation, particularly since the present model does not directly represent a current real-world topology from which practical experience would have provided pertinent lessons. This general objective, namely "insight production", proved to be a time consuming, and expensive (from a CPU utilization point of view) effort.

One of the most interesting observations was that of apparent *network gridlock*. By gridlock it is meant that congestion could reach such an extreme level that messages would no longer get from the source to the destination. In fact, packets would seem to remain in a state of near perpetual retransmission, PSN-to-PSN. Gridlock seemed to propagate when traffic levels (both originals and retransmissions) reached a point such that as buffers in front of PSN's filled, causing the loss of large numbers of packets, PSN-PSN links would remain occupied awaiting retransmission timeouts. This resulted in increased utilization of links, until such time as one PSN would have all its outgoing links in use, which caused its server to stop processing. This caused its queue to be constantly filled, which propagated the backlog to the adjacent nodes. This phenomenon is, most likely, a result of the topology of the network: it has limited alternate routes between destinations. In addition the routing is fixed for the ten second interval between

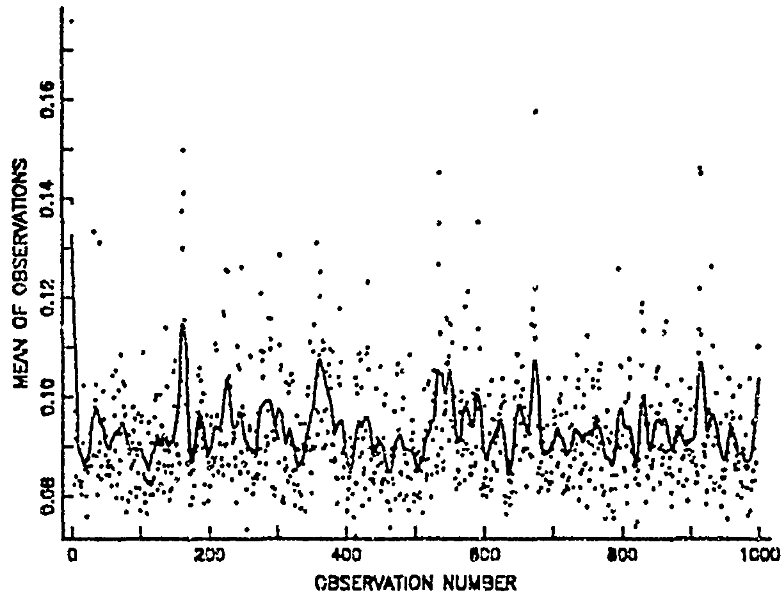

Figure 8. Determination of the Length of the Transient Phase

routing updates. It is likely that the DDN avoids this by the proper implementation of the congestion control function.

Gridlock was initially observed as host-to-host timeout intervals were decreased beyond a specific point for various traffic generation levels; the increased retransmission rates clearly contributed to the overall congestion on the network. In this situation the congestion which lead to gridlock appeared to be caused by the excessive number of unnecessary retransmissions induced by the very short timeout interval. This unnecessary traffic put increased demand on network resources and would eventually lead to the gridlock described above. This increased congestion and eventual gridlock seems to ex-

plain the results discussed earlier, where the response time was observed to increase rapidly as host-to-host timeouts were reduced.

In simulation runs for which the receiving host's buffer space is limited, so as to induce end-to-end losses, the gridlock phenomenon was also observed at high host-to-host timeout intervals. In this situation the excessive congestion which leads to gridlock appeared to result from buffer space being held for an inordinate length of time while awaiting the legitimate retransmission of a message which had packets destroyed in transit. Because this message holds the reservation for a "long" period of time it causes a limited resource (buffer space) to become scarcer. This results in packets of other messages being unable to make a reservation, resulting in their eventual retransmission. In this manner traffic (which includes both new originals and many retransmissions) builds up on the network until it reaches a level which induces the gridlock phenomenon described above. DDN avoids this scenario by use of an adaptive sliding window which would result in a sending host "shutting" the window upon notification (by means of an EACK) that the receiver has no buffer space for additional new messages. Once buffer space becomes available, the window would again open to allow the flow of new messages to that destination host.

E. EFFECT OF CHANGES IN PACKET SWITCHING NODE TIMEOUTS

An experiment was conducted to determine the effect of varying the timeout interval for packet switching node (PSN) retransmissions; that is, the effect of the time that is allowed to elapse between the transmission of a packet (PSN-to-PSN) and a retransmission of that same packet if no acknowledgement is received.

For this experiment two major aspects of the simulation were modified. The first was that buffer space was set at all hosts so that there is no loss of packets at the receiving host as a result of a packet being unable to make a reservation for its entire message. This was done to identify the changes in response time with the changes in the PSN timeout interval.

The second change was that 100% of each source host's message traffic was destined to the receiving host directly opposite it in the topology (i.e., A to D, B to C, C to B, D to A). This was done so as to increase the amount of traffic throughout the network in order to force conflicting demands on network resources, in particular, PSN resources.

The other parameters of the simulation (with the exception of node timeout interval) were fixed for all runs of the experiment. Simulation runs were for 500 simulation seconds, which resulted in 18 batches for estimation of the mean response time. Batch sizes

were determined in the manner discussed earlier and remained at 100 observations per batch. Each simulation run utilized the same random number generator seeds. The node timeout interval was varied from 0.0012 to 0.085 seconds. The range of the timeout interval changes was limited to values greater than the PSN-to-PSN propagation delay and thus the choice of 0.0012 as the lower limit on the range. This is forced by the logical flow of entities (packets) through the simulation code. This logical flow would not allow a retransmission to occur before the original propagation delay had expired without causing an execution time error. The upper limit on the interval (0.085) was chosen after observing the data and noting that the behavior of the performance seemed to be well defined by this point.

An estimate of the mean response time was formed from the batch means of each run. These results are plotted in Figure 9 on page 38. The actual batch observations along with the estimates formed and their confidence intervals are reported in Appendix C. It was suggested that a curve of the form $y = a + \frac{b}{x} + c \times x$, be fitted with x being the node timeout interval and y the estimate of the mean response time. This fitted curve is also shown in Figure 9 on page 38.

Table 1. NODE TIMEOUT INTERVAL EXPERIMENT TABLE OF COEFFICIENTS 1,

R-SQUARED = 0.96267			STANDARD ERROR = 0.00046929		
ADJ R-SQUARED = 0.9608					
				0.95 CONFIDENCE LIMITS	
COEFF	ESTI-MATE	STD ERR	SIG LEVEL	LOWER	UPPER
A	9.5824E-2	1.7121E-4	2.7756E-17	9.547E-2	9.6170E-2
B	2.3371E-7	3.8748E-7	5.4982E-1	-5.4981E-7	1.0172E-6
C	9.2656E-2	3.7417E-3	8.3267E-17	8.5090E-2	1.0022E-1

The table of coefficients (Table 1) shows the results of this fit. These results show that the adjusted R-squared value of 0.9608 implies a good fit. It should also be noted that the b coefficient is insignificant in this fit (at the $\alpha = 0.05$ level). This suggests that dependence of mean response time on very small values of node timeout interval is slight, or that timeout interval can safely be made very small at little cost. A reason for this behavior is discussed later. It should also be noted that because common random

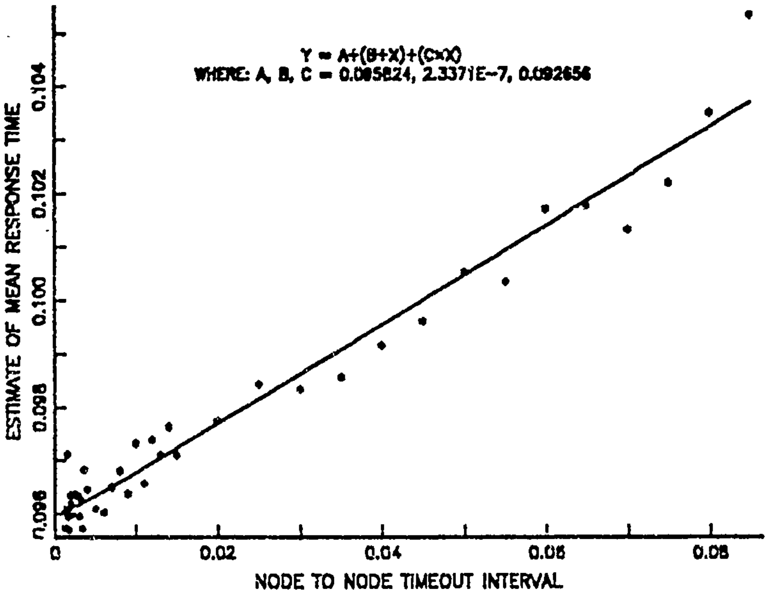

Figure 9. **Estimate of Response Time at Various Node Timeout Intervals**

number generator seeds are used throughout this experiment statistical inference from this fit may be inappropriate and more properly restricted to developing a "feel" for the coefficients.

Figure 10 on page 39 shows the same data and fitted curve, as seen earlier, but only in the timeout interval range 0.001 to 0.010. Of interest here is that the fitted curve flattens out at shorter timeout intervals.

The performance expected was that an increase in mean response time would eventually result if the timeout intervals were increased (as is observed), a minimum value would occur at some timeout somewhat greater then the round trip time, and an increase in mean response time would again occur at very short timeout intervals. The increase

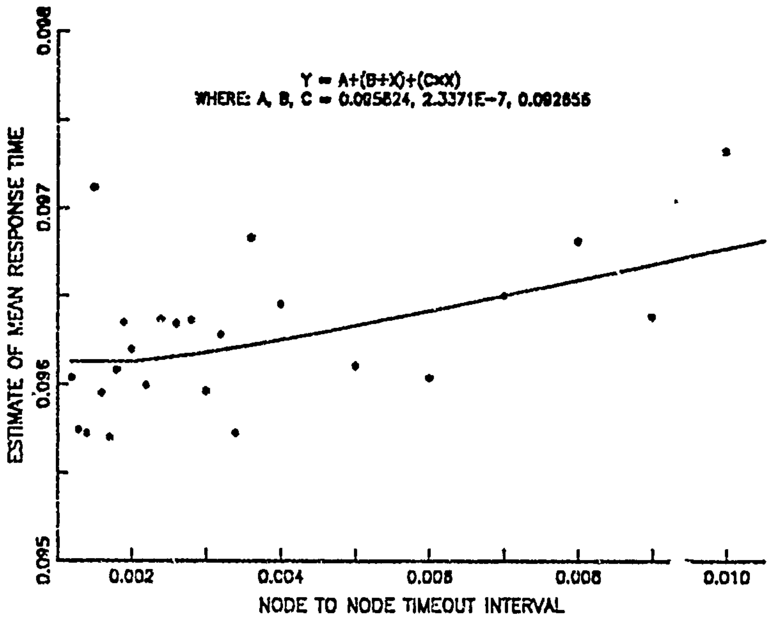

Figure 10. Estimate of Response Time Node Experiment (Expanded Scale)

at very short timeout intervals was expected as a result of competition for processor buffer space which would cause packets to be lost when that buffer space became full.

The observed performance, though not anticipated, becomes plausible when compared, in detail, to protocol implementation. Upon transmission of a packet to an adjacent PSN and after a negligible time delay to transit the transmission line, the receiving PSN's link control (LC) checks to ensure the packet is not a duplicate, discarding the packet if it is a duplicate. LC then performs a checksum for errors (discarding the packet if it is in error). The packet is then sent to store and forward (S&F) for further processing. This is where some form of acknowledgement (ACK) is generated. Since the ACK is appended to the next outgoing packet (routed to the appropriate PSN) or a null

packet generated after a minimal delay (i.e., ACK's are given priority handling by S&F) the delay in transmitting an ACK is negligible as compared to the delay involved with the actual forwarding of the packet. Because of this and because LC gives priority handling to incoming packets, S&F is where the packet first encounters buffer limitations, and thus, competes for resources. If the buffer space is limited (full) due to high traffic levels, then those packets which can not enter the buffer are discarded. In this situation the more often a packet attempts to enter the buffer (the shorter the timeout interval) the more likely it is to be able to enter the buffer. In any situation, it is unreasonable to set the timeout interval at any value less then some estimate of the round trip time, PSN-to-PSN. Where the round trip time is the total time for the packet to propagate along the transmission lines, be processed by LC and the acknowledgement to propagate back to the sending node. This is true because it is an obvious waste of resources to retransmit a packet before the acknowledgement has a reasonable chance to arrive. If the estimate of round trip time is short, it will result in any unnecessarily retransmitted packets (duplicates) being destroyed at the destination node but an increased number of attempts to enter the crowded buffers for those legitimate retransmissions and thus improved probability of entry. As the timeout interval increases to the actual roundtrip time, the number of unnecessary retransmitted packets drops to zero, but there are fewer attempts to enter the buffer by the legitimate retransmittals. If the timeout interval is increased further, any packet awaiting retransmission simply passes time waiting to be retransmitted and further increases the time it takes to complete the entry into the buffer.

This logic implies that the estimated mean response time should increase as the node-to-node timeout interval increases; this is observed in Figure 9 on page 38.

Table 2. NODE TIMEOUT INTERVAL EXPERIMENT TABLE OF COEFFICIENTS 2.

R-SQUARED = 0.96233			STANDARD ERROR = 0.00046563		
ADJ R-SQUARED = 0.96141					
				0.95 CONFIDENCE LIMITS	
COEFF	ESTIMATE	STD ERR	SIG LEVEL	LOWER	UPPER
A	0.095911	0.000091743	8.3267E-17	0.095725	0.096096
B	0.091187	0.0028174	2.2204E-16	0.085496	0.096877

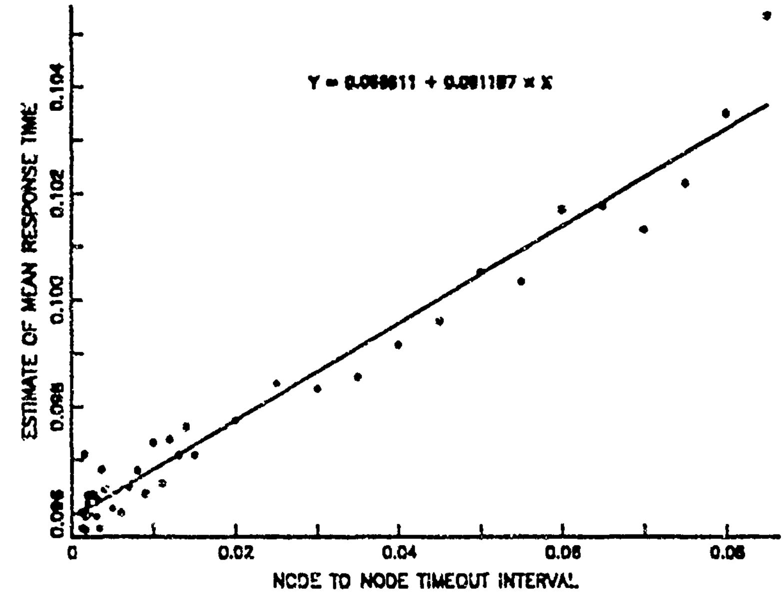

Figure 11. Estimated Mean Response Time Node Timeout Experiment

This analysis is further highlighted by again fitting a curve to the data, however in this case it is a simple linear fit (see Figure 11). The table of coefficients (Table 2 on page 40) shows the results of this fit. Note that all coefficients are significant and the adjusted R^2 value of 0.96141 shows a slight improvement to that of the previous fit.

The previous experiment was run with traffic origination levels set at moderately high levels (approximately 3 messages per second) so that there was no excessive demand on network resources, particularly PSN buffer space.

An additional experiment was run at traffic levels (approximately four messages per second) such that PSN buffers were observed to fill completely during the simulation. The estimates of mean response time are shown in Figure 12 on page 43. The actual batch observations along with the estimates formed and their confidence intervals are reported in Appendix C. The data points marked with a G indicate simulation runs during which the simulation entered gridlock prior to the completion of the run. In this case the gridlock seemed to result from having the links between PSN's being occupied for a "long" period of time while awaiting the retransmission of a packet that did not enter the next PSN on its route (for whatever reason). This caused the PSN buffer to fill more often or stay completely full propagating the congestion around the network.

Also of interest here is that the data from this experiment seems to have separated into 2 subsets, high (mean response time greater then 0.1035) and low (mean response time less than 0.1035) as seen in Figure 12 on page 43. Both subsets display the same increasing response time as timeout interval increases, as was observed in the previous experiment. An explanation for the breakdown into two subsets might be that because traffic levels were set so that buffers were filled periodically (and thus packets lost when full buffers were encountered) various timeout intervals might induce overlapping of any high traffic density periods resulting from the nature of the traffic origination scheme. This would cause an even greater load on the buffer space (similar to a harmonic frequency). This effect was further exacerbated at the very high timeout intervals, for which the load became so high that the network entered gridlock.

A comparison of these two experiments is displayed in Figure 13 on page 44; there it is clearly seen that there is an increase in mean response time as traffic loads increase as well as the increase due to the increase in timeout interval.

F. EFFECT OF CHANGES IN HOST TIMEOUT VALUES

An experiment was conducted to investigate the effect of changing host retransmission timeout intervals upon the mean response time. In this experiment the host buffer space was reduced at host D. Additionally some p_{ij} fraction of the traffic from hosts B and C was randomly destined to host D. All of host A's traffic was destined to host D. These conditions were established in order to cause conflicting demands on limited buffer resources at host D. This will force some percentage of the host A to

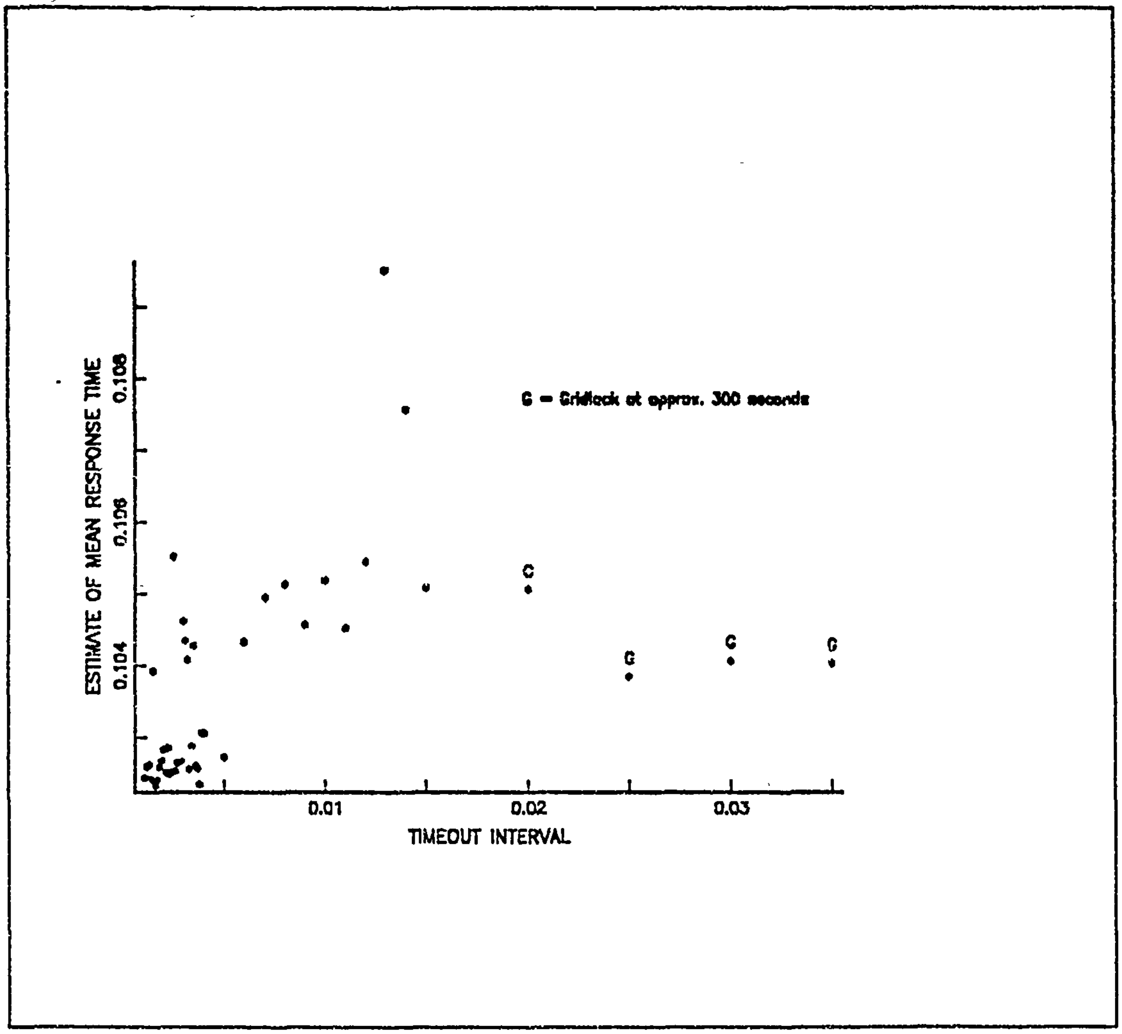

Figure 12. Estimates of Mean Response Time at High Traffic Levels

host D packets attempting to make buffer reservations to be destroyed for lack of sufficient buffer space. In all simulation runs the remaining parameters were fixed with the exception of the host timeout intervals. Parameters were fixed as described in Chapter 4 with $p_{A,D} = 1.0$, $p_{B,A} = p_{C,A} = 0.15$, $p_{B,C} = p_{C,B} = 0.60$, $p_{B,D} = p_{C,D} = 0.25$, $p_{D,A} = p_{D,B} = p_{D,C} = 0.25$, and a message arrival rate of approximately four messages per second. The host timeout interval was varied from 0.30 seconds to 0.62 seconds (with all hosts using the same timeout interval value). This range of values was chosen because, at the rate of message generation used in this experiment, gridlock was encountered at either extreme in the manners described earlier in this section. Each simulation run was executed for 500 simulation seconds, which resulted in 18-19 batches for use in

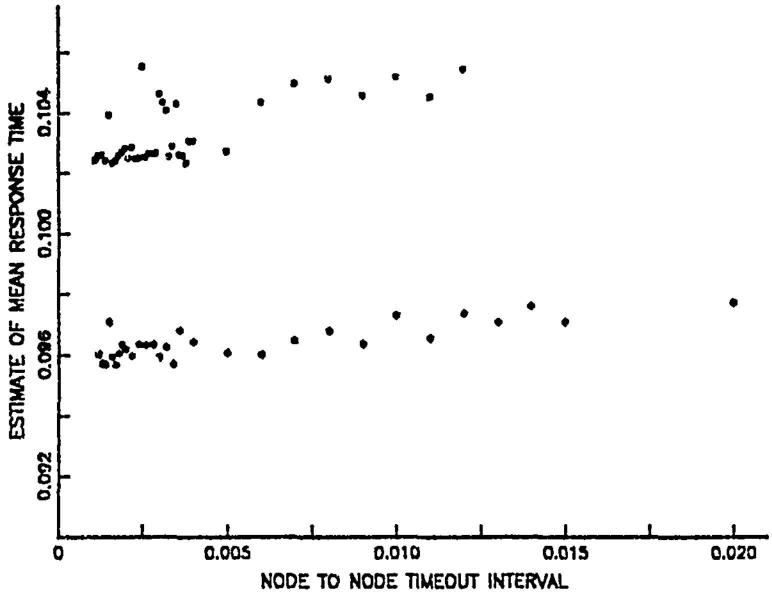

Figure 13. Comparison of Mean Response Time at Two Traffic Density Levels

estimating the mean response time. As discussed earlier, all simulation runs utilized the same random number generator seeds.

An estimate of the mean response time was formed from the batch means of each simulation run and are shown in Figure 14 on page 45 along with a curve fitted to the data. The actual batch observations along with the estimates formed and their confidence intervals are reported in Appendix C.

These data indicate that there is a region for host timeout values at which a host may "optimize" the expected response time for a message completion. Choice of a near optimal timeout is a primary objective in a real analysis.

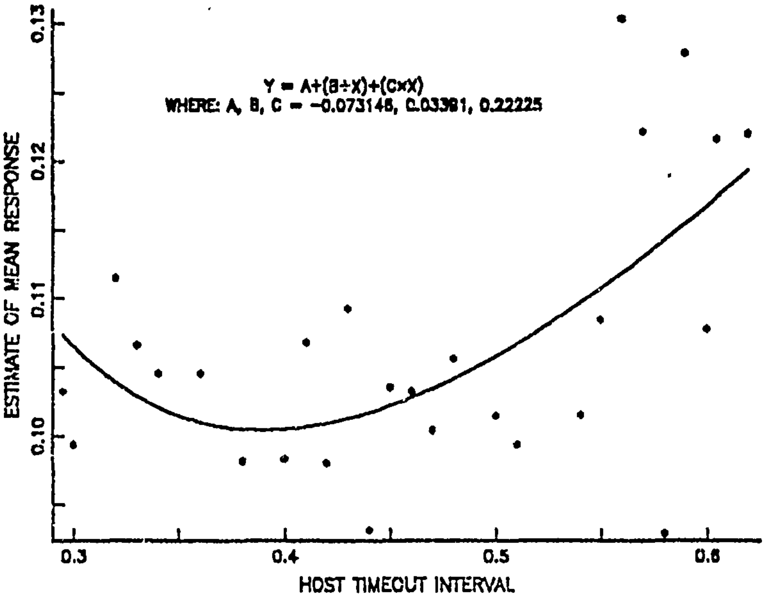

Figure 14. Estimated Mean Response Time at Various Host Timeout Intervals

At shorter timeout intervals, the increase in response time seems to be the result of increased congestion caused by unnecessary retransmissions of messages. This is supported by noting that gridlock is encountered at timeout intervals shorter then those displayed. The upturn at longer timeout intervals can be explained by the longer time spent waiting for retransmission for those messages legitimately requiring retransmission, particularly if it requires more than one retransmission. It should be noted that, at timeout intervals longer than those displayed, the simulation encountered gridlock.

The fitted curve is of the form $y = a + \frac{b}{x} + c \times x$ (where y is the estimated mean response and x is the host timeout interval) and the results of this fit are displayed in

Table 3. Although this fit does not have an R^2 value which would indicate a good fit or the best prediction characteristics, the curve still serves to give a "feel" for what the mean response time is doing over this range of host timeout interval values.

Table 3. TABLE OF COEFFICIENTS FOR HOST TIMEOUT EXPERIMENT

R-SQUARED = 0.36037			STANDARD ERROR = 0.0082752		
ADJ R-SQUARED = 0.30707					
				0.95 CONFIDENCE LIMITS	
COEF	ESTI-MATE	STD ERR	SIG LEVEL	LOWER	UPPER
A	-0.073146	0.069498	0.30306	-0.2166	0.070306
B	0.03391	0.014766	0.030679	0.0034316	0.064389
C	0.22225	0.078423	0.0091738	0.060377	0.38412

Selected box plots of the batch means which form the estimate at each timeout interval value are shown in Figure 15 on page 47. (All box plots are shown in Appendix D.) These box plots indicate that the increase in estimated mean response, at each end of the range of values investigated, is traceable to a few batches with very high mean response times rather than to a general tendency for all batch means to be increasing; that is, the simulation runs at the high and low timeout intervals suggest that response times have greater variance and positive skewness then those in the mid-range timeout intervals. However, it appears that this variance can be minimized by operating in the area of the minimum of the curve fitted earlier. At this point the network will ordinarily operate in such a way that all packets are likely to proceed efficiently through the network.

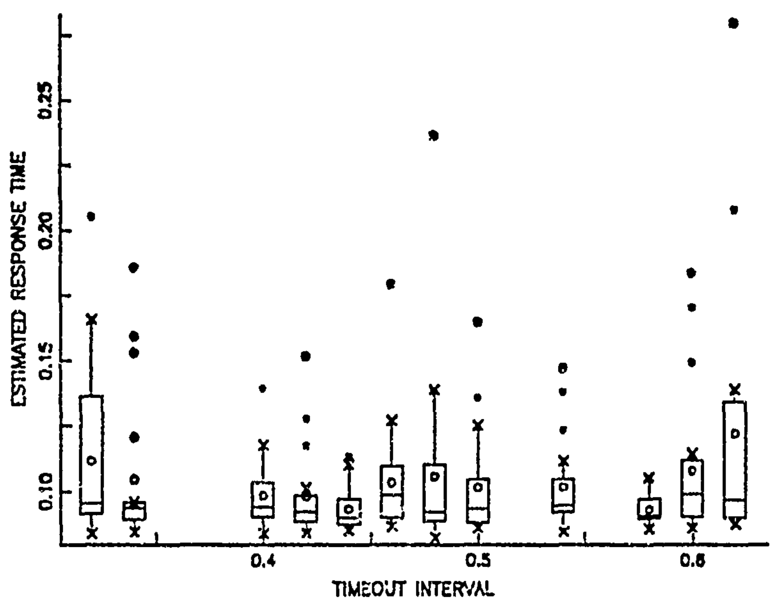

Figure 15. Selected Box Plots for the Host Timeout Experiment

VI. SUMMARY

A. CONCLUSIONS

The most significant conclusion of this thesis is that the simulation model developed here is capable of providing a valuable understanding of the dynamics and performance of a data communications network. One shortfall is that the model exhibits gridlock which inhibits comparison directly to DDN performance which, presently, does not exhibit gridlock. A less significant shortfall is that the computer execution time seems to be longer then desirable.

The analysis of output has demonstrated that the timeout interval (in both experiments) directly affects the performance of the network. In the case of host timeout interval it has been shown that there is a region of timeout intervals which will generally tend to cause the most satisfactory network performance overall; that is, that being "greedy" (very short timeout intervals) or "cautious" (very long timeout intervals) regarding the setting of the host timeout interval, will reduce network performance. In the case of node timeout intervals, it was shown that being "greedy" will slightly improve network performance. However, this will be an inefficient use of resources in that the transmission device will be utilized for the unnecessary retransmission of packets. Thus the best place to set node timeout interval is at some value slightly larger then the estimate of the roundtrip time between the PSN's plus intra-PSN processing time.

B. AREAS FOR FURTHER STUDY

During analysis of the simulation model and data output several areas for further research were identified. They include both possible improvements to the simulation model and areas for further performance analysis.

The first area for improvement in the simulation model should be in the routing algorithm. As discussed, routing is updated every ten seconds. However, the DDN routing function tracks delays between adjacent PSN's and if the delay exceeds some threshold a routing update is issued. An implementation of this, on a very minimal level, should greatly reduce the gridlock phenomenon observed.

Another area for improvement would be to eliminate the detailed processing at the receiving hosts for hosts A,B, and C. This should be appropriate because the traffic of interest was from host A to host D. The remaining inter-host traffic was modeled to put conflicting demands on network resources. By eliminating the full processing at hosts

A,B, and C while retaining any associated processing time delays simulation execution time should significantly improve without loss of generality.

Elimination of full processing at hosts A, B, and C will allow expansion of the topology without adversely affecting available computing resources or involving excessive execution times. If so implemented a suggested topology would be to have PSN's arranged at the vertices of a hexagon with another PSN centered within the hexagon. This would allow multiple routes, of differing lengths, between hosts. The modular setup of the current simulation model should make it easily expandable.

One area of performance analysis which may be of particular interest would be to study response time as a function of host timeout interval and traffic density. This may identify whether the optimal setting of host timeout interval varies significantly under different traffic loads or if a single timeout value is sufficient for all loads; this latter suggestion seems unlikely. Another area of investigation would be to examine the effect of the routing table update interval upon system performance. A third area of interest is to study the individual routing times observed in the host timeout experiment to determine if there is any indicator that could be used to predict the onset of poor response time periods. If so, then possible corrective action could be developed such as throttling host traffic. This can not be done with a batch means approach and would require multiple runs of long duration.

C. SUMMARY

This subject area will become increasingly more important as more demand is put on the limited resources within the DDN. In the environment of reduced budgets, it is not likely that money will be available to solve capacity or performance problems. Thus continued studies such this should provide valuable insight in ways to better utilize limited resources.

APPENDIX A. SIMULATION MODEL CODE

This appendix includes the model code used in this thesis. The code was written in SLAM II Version 4.03 (copyright 1983, Pritsker and Associates, Inc.) for use on an IBM mainframe computer model 3033AP with VS FORTRAN V2 compiler (Naval Postgraduate School Mainframe computer).

A. SLAM II CODE

```
GEN,CDR J KIRWAN,OA THESIS,4/22/1990,1,NO,NO,YES/YES,NO,YES/1,72;
LIMITS,65,9,10000;
;      INITIALIZE THE RANDOM NUMBER GENERATOR SEEDS
SEEDS,267891(1)/Y,220605(2)/Y,656423(3)/Y,781173(4)/Y,478923(10)/Y;
SEEDS,363893(5)/Y,3201605(6)/Y,95645523(7)/Y,2851173(8)/Y, 57646551(9)/Y;
;      INITIALIZE THE MESSAGE SERIAL NUMBERS FOR HOSTS A,B,C,D
INTLC,XX(1)=10000,XX(2)=20000,XX(3)=30000,XX(4)=40000;MSG SERIAL NUMBERS
;      INITIALIZE THE MEAN OF THE MESSAGE INTERARRIVAL TIME DISTRIBUTION
INTLC,XX(31)=.26,XX(32)=.26,XX(33)=.26,XX(34)=.26;
;      INITIALIZE THE "PROPAGATION" DELAY BETWEEN PSN'S 1,2,3,4,5
INTLC,XX(41)=.001,XX(42)=.001,XX(43)=.001,XX(44)=.001,XX(45)=.001;
;      INITIALIZE THE HOST TIMEOUT INTERVAL AT HOSTS A,B,C,D
INTLC,XX(51)=.49,XX(52)=.49,XX(53)=.49,XX(54)=.49;
;      INITIALIZE THE NODE TIMEOUT INTERVAL AT NODES 1,2,3,4,5
INTLC,XX(61)=.002,XX(62)=.002,XX(63)=.002,
      XX(64)=.002,XX(65)=.002;
;      INITIALIZE THE HOST RECEIVING SIDE BUFFER SPACE
;      (i.e. 10= room for 10 messages at the receiving side of the host
INTLC,XX(96)=10,XX(97)=10,XX(98)=10,XX(99)=2;
;      INITIALIZE STATISTICAL GLOBALS
INTLC,XX(91)=0,XX(92)=0,XX(93)=0;
;      INITIALIZE THE FRACTION FROM ANY HOST TO ANY DESTINATION
;INTLC,XX(12)=0.0,XX(13)=0.0;   FROM HOST A TO HOST B,C, BALANCE TO D
;INTLC,XX(14)=.15,XX(15)=.60;   FROM HOST B TO HOST A,C, BALANCE TO D
;INTLC,XX(16)=.15,XX(17)=.60;   FROM HOST C TO HOST A,B, BALANCE TO D
;INTLC,XX(18)=.25,XX(19)=.25;   FROM HOST D TO HOST A,B, BALANCE TO C
;      INITIALIZE THE NODE PROCESSOR TIME DELAY AT NODES 1,2,3,4,5
INTLC,XX(25)=.012,XX(26)=.012,XX(27)=.012,
      XX(28)=.012,XX(29)=.012;
;
;      INITIALIZE THE ROUTING TABLE
ARRAY(1,4)/0,2,4,2;
ARRAY(2,4)/1,0,3,5;
ARRAY(3,4)/1,2,4,5;
ARRAY(4,4)/1,1,0,5;
ARRAY(5,4)/4,2,4,0;
```

```
     NETWORK;
;
     RESOURCE/1,NODE1R(48),61;                LIMITS THE NUMBER OF
     RESOURCE/2,NODE2R(48),62;                PACKETS ANY NODE CAN
     RESOURCE/3,NODE3R(64),63;                HAVE ON OUTGOING
     RESOURCE/4,NODE4R(48),64;                LINKS.
     RESOURCE/5,NODE5R(48),65;

     RESOURCE/WINDOWA(75),1;                LIMITS THE NUMBER OF
     RESOURCE/WINDOWB(75),2;                PACKETS ANY HOST CAN
     RESOURCE/WINDOWC(75),3;                HAVE ON SUBNET   APPROXS
     RESOURCE/WINDOWD(75),4;                A SLIDING WINDOW
;
; SET UP HOST A INPUT NETWORK
;
;
HSTA    CREATE,EXPON(XX(31),1),0,1,,1;                      CREATE MSGS
;              ASSIGN INITIAL ATTRIBUTES TO THE MESSAGE
;              ATRIB(1)= MESSAGE ENTRY TO SUBNET TIME
;              ATRIB(2)= PLACEHOLDER USED IN ROUTE
;              ATRIB(3)= ORGIN HOST FOR THIS MESSAGE
;              ATRIB(4)= DESTINATION HOST FOR THIS MESSAGE
;              ATRIB(5)= CURRENT NODE LOCATION FOR THIS MESSAGE
;              ATRIB(6)= MESSAGE SERIAL NUMBER
;              ATRIB(7)= PACKETS PER MESSAGE
;              ATRIB(8)= PACKET NUMBER (PACKET XX OF MESSAGE YY)
;              ATRIB(9)= RETRANSMISSION COUNTER
      ASSIGN,ATRIB(3)=1,
           ATRIB(4)=4,
           ATRIB(5)=0,
           ATRIB(6)=XX(1),
           XX(1)=XX(1)+1,
           ATRIB(7)=3,
           ATRIB(9)=0,1;
      GOON,1;
      ACT,,XX(12),AB;
      ACT,,XX(13),AC;
      ACT,,,ADST;
AB    ASSIGN,ATRIB(4)=2,1;
      ACT,,,ADST;
AC    ASSIGN,ATRIB(4)=3,1;
      ACT,,,ADST;
ADST  GOON,3;
      ACT,,,A1;
      ACT,,,A2;
      ACT;
        ASSIGN,ATRIB(8)=3,1;                         3 PACKETS PER MSG
        ACT,,,RESA;
A1    ASSIGN,ATRIB(8)=1,1;
        ACT,,,RESA;
A2    ASSIGN,ATRIB(8)=2,1;
;
RESA  AWAIT(1/600),WINDOWA/1,BALK(DMPA),1;   SLIDING WINDOW BUFFER
      ASSIGN,ATRIB(1)=TNOW,2;
      ACT,,,AA;
```

```
      ACT;
;*************************     RETRANSMIT LOOP
AA1   EVENT,7,1;                PUT THE PACKET IN THE OKAY TO RETX FILE
RTXA  ASSIGN,ATRIB(9)=ATRIB(9)+1,1;       INCREMENT RETX COUNT
      ACT,XX(51),,;                          RETX TIMEOUT
T11   EVENT,4,1;                          TEST IF OKAY TO RETX
      ACT,,XX(5) .EQ. 0,DMP1;              IF NOT - DUMP IT
      ACT;                                 IF OKAY - SEND IT
T12   ASSIGN,XX(95)=XX(95)+1,2;
      ACT,,,RTXA;                         LOOP FOR NEXT RETX
      ACT,,,NDE1;                         SEND THIS RETX TO SUBNET
;*************************
AA    QUEUE(19),,,,;                    HOLDING QUEUE IF PSN BUFFER FULL
      ACT,,,HN1;
;
DMPA  ASSIGN,XX(71)=XX(71)+1;
      TERM;
;
; SET UP HOST B INPUT NETWORK
;
;
HSTB  CREATE,EXPON(XX(32),2),0,1,,1;
      ASSIGN,ATRIB(3)=2,
        ATRIB(4)=3,
        ATRIB(5)=0,
        ATRIB(6)=XX(2),
        XX(2)=XX(2)+1,
        ATRIB(7)=3,
        ATRIB(9)=0,1;
      GOON,1;
      ACT,,XX(14),BA;
      ACT,,XX(15),BD;
      ACT,,,BDST;
BA    ASSIGN,ATRIB(4)=1;
      ACT,,,BDST;
BD    ASSIGN,ATRIB(4)=4;
      ACT,,,BDST;
BDST  GOON,3;
      ACT,,,B1;
      ACT,,,B2;
      ACT;
      ASSIGN,ATRIB(8)=3;
      ACT,,,RESB;
B1    ASSIGN,ATRIB(8)=1;
      ACT,,,RESB;
B2    ASSIGN,ATRIB(8)=2;
;
RESB  AWAIT(2/600),WINDOWB/1,BALK(DMPB),2;
      ACT,,,BB;
      ACT;                                  RETRANSMIT LOOP
BB1   EVENT,7,1;
RTXB  ASSIGN,ATRIB(9)=ATRIB(9)+1,1;       INCREMENT RETX COUNT
      ACT,XX(52);
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP2;
      ACT;
```

```
      GOON,2;
      ACT,,,RTXB;                          LOOP FOR NEXT RETX
      ACT,,,NDE2;                           SEND THIS RETX TO SUBNET
BB    QUEUE(29),,,,;
      ACT,,,HN2;
;
DMPB  ASSIGN,XX(72)=XX(72)+1;
      TERM;
;
;
; SET UP HOST C INPUT NETWORK
;
;
HSTC  CREATE,EXPON(XX(33),3),0,1,,1;
      ASSIGN,ATRIB(3)=3,
        ATRIB(4)=2,
        ATRIB(5)=0,
        ATRIB(6)=XX(3),
        XX(3)=XX(3)+1,
        ATRIB(7)=3,
        ATRIB(9)=0,1;
      GOON,1;
      ACT,,XX(16),CA;
      ACT,,XX(17),CD;
      ACT,,,CDST;
CA    ASSIGN,ATRIB(4)=1;
      ACT,,,CDST;
CD    ASSIGN,ATRIB(4)=4;
      ACT,,,CDST;
CDST  GOON,3;
      ACT,,,C1;
      ACT,,,C2;
      ACT;
      ASSIGN,ATRIB(8)=3;
      ACT,,,RESC;
C1    ASSIGN,ATRIB(8)=1;
      ACT,,,RESC;
C2    ASSIGN,ATRIB(8)=2;
;
RESC  AWAIT(3/600),WINDOWC/1,BALK(DMPC),2;
      ACT,,,CC;
      ACT;                                  RETRANSMIT LOOP
CC1   EVENT,7,1;
RTXC  ASSIGN,ATRIB(9)=ATRIB(9)+1,1;       INCREMENT RETX COUNT
      ACT,XX(53),,;
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP4;
      ACT;
      GOON,2;
      ACT,,,RTXC;                          LOOP FOR NEXT RETX
      ACT,,,NDE4;                           SEND THIS RETX TO SUBNET
CC    QUEUE(49),,,,;
      ACT,,,HN4;
;
DMPC  ASSIGN,XX(73)=XX(73)+1;
```

```
     TERM;
;
;
; SET UP HOST D INPUT NETWORK
;
;
HSTD  CREATE,EXPON(XX(34),4),0,1,,1;
      ASSIGN,ATRIB(3)=4,
        ATRIB(4)=1,
        ATRIB(5)=0,
        ATRIB(6)=XX(4),
        XX(4)=XX(4)+1,
        ATRIB(7)=3,
        ATRIB(9)=0,1;
      GOON,1;
      ACT,,XX(18),DB;
      ACT,,XX(19),DC;
      ACT,,,DDST;
DB    ASSIGN,ATRIB(4)=2;
      ACT,,,DDST;
DC    ASSIGN,ATRIB(4)=3;
      ACT,,,DDST;
DDST  GOON,3;
      ACT,,,D1;
      ACT,,,D2;
      ACT;
       ASSIGN,ATRIB(8)=3;
       ACT,,,RESD;
D1    ASSIGN,ATRIB(8)=1;
       ACT,,,RESD;
D2    ASSIGN,ATRIB(8)=2;
;
RESD  AWAIT(4/600),WINDOWD/1,BALK(DMPD),2;
      ACT,,,DD;
      ACT;                                    RETRANSMIT LOOP
DD1   EVENT,7,1;
RTXD  ASSIGN,ATRIB(9)=ATRIB(9)+1,1;
      ACT,XX(54),,;
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP5;
      ACT;
      GOON,2;
      ACT,,,RTXD;
      ACT,,,NDE5;
DD    QUEUE(59),,,,;
      ACT,,,HN5;
;
DMPD  ASSIGN,XX(74)=XX(74)+1;
      TERM;
;-----------------------------------------------------------------
;
;SET UP PACKETING SWITCHING NODE 1
;
HN1   QUEUE(11),,10,BLOCK,SL10;              DEDICATED BUFFER FOR HOST
NDE1  QUEUE(12),,10,BALK(DMP1),SL10;         BUFFER FOR NODE TO NODE USE
;
```

```
DMP1  TERM;
;
SL10  SELECT,CYC,,,HN1,NDE1;          SERVICE ALTERNATES BETWEEN QUEUES
      ACT,XX(25);                      NODE PROCESSOR
      AWAIT(61/1),NODE1R/1,BLOCK,1;   LIMITS NODE TO NODE LINKS
T1    ASSIGN,ATRIB(2)=ATRIB(5),
      ATRIB(5)=1;                     SET INDICATORS: WHERE IT CAME FROM,
;                                     WHERE IT IS
T3    EVENT,6,1;                      ADAPTIVE ROUTING
T4    ASSIGN,ATRIB(2)=XX(11),1;       SET WHERE IT WAS ROUTED TO
RL1   GOON,2;                         DUPLICATE PACKETS AND
      ACT,,,RTX1;                          SEND ONE TO RETX LOOP
      ACT;                                 AND ONE TO TRANSMISSION
      GOON,1;
      ACT,XX(41),.01,DMP1;            TRANSMISSION DELAY WITH ERROR
      ACT,XX(41);                     TRANSMISSION DELAY WITH NO ERROR
      EVENT,8,1;
      ACT,,ATRIB(2) .EQ.  2 ,NDE2;    AND 1 TO SUBNET NODE 2
      ACT,,ATRIB(2) .EQ.  3 ,NDE3;          OR TO SUBNET NODE 3
      ACT,,ATRIB(2) .EQ.  4 ,NDE4;          OR TO SUBNET NODE 4
      ACT,,ATRIB(2) .EQ.  0 ,RECA;          OR TO RECEIVING HOST
      ACT,,,DMP1;
; *****************          RETX LOOP
RTX1  GOON;
      ACT,XX(61);                      DELAY FOR NODE TIMEOUT
      EVENT,9,1;                       CHECK IF RETX REQUIRED
      ACT,,XX(5) .EQ. 0,DMP1;          IF NOT DESTROY PACKET
      ACT,,,RL1;                       LOOP TO TRANSMIT POINT
;
;
;*****************  RECEIVING SIDE OF HOST
RECA  ASSIGN,ATRIB(5)=9,1;
;                   IF MSG COMPLETED DESTROY PACKET
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP1;
      ACT;
;                 IF BUFFER RESERVED FOR MSG JUMP TO SORTING PACKETS
      EVENT,3,1;
      ACT,,XX(5) .NE. 0,SPA;
;                 IF NEITHER OF THE ABOVE TRY TO RESERVE BUFFER FOR MSG
      ACT;
;
RSVA  GOON,1;
      ACT,,XX(96) .LT. 1 ,DMP1;         IF NO ROOM FOR MSG DELETE PKT
      ACT,,XX(96) .GE. 1;               IF ROOM FOR MSG RESERVE IT
      ASSIGN,XX(96)=XX(96)-1;
      EVENT,1,1;
;
;
SPA   GOON,1;
      ACT,,ATRIB(8) .EQ. 1,SA1;              SORT BY PKT NUMBER
      ACT,,ATRIB(8) .EQ. 2,SA2;
      ACT,,ATRIB(8) .EQ. 3,SA3;
;
;  CHECK TO SEE IF PACKET ALREADY HERE FOR THIS MSG  IF YES DESTROY
SA1   EVENT,5,1;
```

```
      ACT,,XX(5) .EQ. 0,QA1;
      ACT,,,DMP1;
SA2   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QA2;
      ACT,,,DMP1;
SA3   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QA3;
      ACT,,,DMP1;
;
QA1   QUEUE(13),,,,MTHA;                        BUFFER UNTIL ALL PKTS OF
QA2   QUEUE(14),,,,MTHA;                        MSG ARRIVE
QA3   QUEUE(15),,,,MTHA;
;
MTHA  MATCH,6,QA1/NA1,QA2/NA1,QA3/NA1;        WHEN ALL PKTS HERE PASS ON
;
NA1   ACCUM,3,3,HIGH(9);
;
      ASSIGN,XX(96)=XX(96)+1;                   MSG IS COMPLETE
      EVENT,2,1;
      ACT,,ATRIB(3) .NE. 4,RA1
      ACT;
NA2   COLCT,INT(1),DTOA ROUTING TIME;15/0/.01,1;
      ASSIGN,XX(24)=TNOW-ATRIB(1),1;          COLLECT THE ROUTE TIME D TO A
RA1   GOON,1;
      ACT,XX(21);                 DELAY THE EACK BY THE ROUTE TIME A TO D
      EVENT,7,1;                    PULL THIS MSG FROM THE RETX FILE
      ACT,,ATRIB(3) .EQ. 2,FRBA;
      ACT,,ATRIB(3) .EQ. 3,FRCA;
      ACT;
AAAA  FREE,WINDOWD/3,1;                        SLIDE THE SLIDING WINDOW
      TERM;
FRBA  FREE,WINDOWB/3,1;                        SLIDE THE SLIDING WINDOW
      TERM;
FRCA  FREE,WINDOWC/3,1;                        SLIDE THE SLIDING WINDOW
      TERM;
;
;
;
;
;SET UP PACKET SWITCHING NODE 2
;
HN2   QUEUE(21),,10,BLOCK,SL20;
NDE2  QUEUE(22),,10,BALK(DMP2),SL20;
;
DMP2  TERM;
;
SL20  SELECT,CYC,,,HN2,NDE2;
      ACT,XX(26);
      AWAIT(62/1),NODE2R/1,BLOCK,1;
      ASSIGN,ATRIB(2)=ATRIB(5),
      ATRIB(5)=2;
      EVENT,6,1;
      ASSIGN,ATRIB(2)=XX(11),1;
RL2   GOON,2;
      ACT,,,RTX2;
      ACT;
```

```
      GOON,1;
      ACT,XX(42),.01,DMP2;
      ACT,XX(42);
      EVENT,8,1;
      ACT,,ATRIB(2) .EQ. 1 ,NDE1;
      ACT,,ATRIB(2) .EQ. 3 ,NDE3;
      ACT,,ATRIB(2) .EQ. 5 ,NDE5;
      ACT,,ATRIB(2) .EQ. 0 ,RECB;
      ACT,,,DMP2;
;
RTX2  GOON;
      ACT,XX(62);                       DELAY FOR NODE TIMEOUT
      EVENT,9,1;                        CHECK IF RETX REQUIRED
      ACT,,XX(5) .EQ. 0,DMP2;             IF NOT DESTROY PACKET
      ACT,,,RL2;                        LOOP TO TRANSMIT POINT
;
;
;
RECB  ASSIGN,ATRIB(5)=9,1;
;                                IF MSG COMPLETED DESTROY PACKET
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP2;
      ACT;
;                 IF BUFFER RESERVED FOR MSG JUMP TO SORTING PACKETS
      EVENT,3,1;
      ACT,,XX(5) .NE. 0,SPB;
;                 IF NEITHER OF THE ABOVE TRY TO RESERVE BUFFER FOR MSG
      ACT;
;
RSVB  GOON,1;
      ACT,,XX(97) .LT. 1 ,DMP2;
      ACT,,XX(97) .GE. 1;
      ASSIGN,XX(97)=XX(97)-1;
      EVENT,1,1;
;
SPB   GOON,1;
      ACT,,ATRIB(8) .EQ. 1,SB1;
      ACT,,ATRIB(8) .EQ. 2,SB2;
      ACT,,ATRIB(8) .EQ. 3,SB3;
;
;  CHECK TO SEE IF PACKET ALREADY HERE FOR THIS MSG  IF YES DESTROY
SB1   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QB1;
      ACT,,,DMP2;
SB2   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QB2;
      ACT,,,DMP2;
SB3   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QB3;
      ACT,,,DMP2;
;
;
QB1   QUEUE(23),,,,MTHB;
QB2   QUEUE(24),,,,MTHB;
QB3   QUEUE(25),,,,MTHB;
;
```

```
MTHB  MATCH,6,QB1/NB1,QB2/NB1,QB3/NB1;
;
NB1   ACCUM,3,3,HIGH(9);
      ASSIGN,XX(97)=XX(97)+1;
      EVENT,2,1;
      ACT,,ATRIB(3) .NE. 3,RB;
      ACT;
NB2   COLCT,INT(1),CTOB ROUTING TIME;15/0/.01,1;
      ASSIGN,XX(23)=TNOW-ATRIB(1),1;
RB    GOON,1;
      ACT,XX(22);
      EVENT,7,1;
      ACT,,ATRIB(3) .EQ. 1,FRAB
      ACT,,ATRIB(3) .EQ. 4,FRDB
      ACT;
BBBB  FREE,WINDOWC/3,1;
      TERM;
FRAB  FREE,WINDOWA/3,1;
      TERM;
FRDB  FREE,WINDOWD/3,1;
      TERM;
;
;
;
;
;SET UP NODE 4
;
HN4   QUEUE(41),,10,BLOCK,SL40;
;
NDE4  QUEUE(42),,10,BALK(DMP4),SL40;
;
DMP4  TERM;
;
SL40  SELECT,CYC,,,HN4,NDE4;
      ACT,XX(28);
      AWAIT(64/1),NODE4R/1,BLOCK,1;
      ASSIGN,ATRIB(2)=ATRIB(5),
      ATRIB(5)=4;
      EVENT,6,1;
      ASSIGN,ATRIB(2)=XX(11),1;
RL4   GOON,2;
      ACT,,,RTX4;
      ACT;
      GOON,1;
      ACT,XX(44),.01,DMP4;
      ACT,XX(44);
      EVENT,8,1;
      ACT,,ATRIB(2) .EQ. 1 ,NDE1;
      ACT,,ATRIB(2) .EQ. 3 ,NDE3;
      ACT,,ATRIB(2) .EQ. 5 ,NDE5;
      ACT,,ATRIB(2) .EQ. 0 ,RECC;
      ACT,,,DMP4;
;
RTX4  GOON;
      ACT,XX(64);
E94   EVENT,9,1;
```

```
      ACT,,XX(5) .EQ. 0,DMP4;
      ACT,,,RL4;
;
;
;
RECC  ASSIGN,ATRIB(5)=9,1;
;                IF MSG COMPLETED DESTROY PACKET
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP4;
      ACT;
;                 IF BUFFER RESERVED FOR MSG JUMP TO SORTING PACKETS
      EVENT,3,1;
      ACT,,XX(5) .NE. 0,SPC;
;                 IF NEITHER OF THE ABOVE TRY TO RESERVE BUFFER FOR MSG
      ACT;
RSVC  GOON,1;
      ACT,,XX(98) .LT. 1 ,DMP4;
      ACT,,XX(98) .GE. 1;
      ASSIGN,XX(98)=XX(98)-1;
      EVENT,1,1;
;
SPC   GOON,1;
      ACT,,ATRIB(8) .EQ. 1,SC1;
      ACT,,ATRIB(8) .EQ. 2,SC2;
      ACT,,ATRIB(8) .EQ. 3,SC3;
;
;  CHECK TO SEE IF PACKET ALREADY HERE FOR THIS MSG  IF YES DESTROY
SC1   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QC1;
      ACT,,,DMP4;
SC2   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QC2;
      ACT,,,DMP4;
SC3   EVENT,5,1;
      ACT,,XX(5) .EQ. 0,QC3;
      ACT,,,DMP4;
;
;
QC1   QUEUE(43),,,,MTHC;
QC2   QUEUE(44),,,,MTHC;
QC3   QUEUE(45),,,,MTHC;
;
MTHC  MATCH,6,QC1/NC1,QC2/NC1,QC3/NC1;
;
NC1   ACCUM,3,3,HIGH(9);
CCC6  ASSIGN,XX(98)=XX(98)+1;
CCC4  EVENT,2,1;
      ACT,,ATRIB(3) .NE. 2,RC;
      ACT;
NC2   COLCT,INT(1),BTOC ROUTING TIME;15/0/.01,1;
CCC3  ASSIGN,XX(22)=TNOW-ATRIB(1),1;
RC    GOON,1;
      ACT,XX(23);
CCC1  EVENT,7,1;
      ACT,,ATRIB(3) .EQ. 1,FRAC;
      ACT,,ATRIB(3) .EQ. 4,FRDC;
```

```
      ACT;
CCCC  FREE,WINDOWB/3,1;
      TERM;
FRAC  FREE,WINDOWA/3,1;
      TERM;
FRDC  FREE,WINDOWD/3,1;
      TERM;
;
;
;
;
;SET UP NODE 5
;
HN5   QUEUE(51),,10,BLOCK,SL50;
;
NDE5  QUEUE(52),,10,BALK(DMP5),SL50;
;
DMP5  TERM;
;
SL50  SELECT,CYC,,,HN5,NDE5;
      ACT,XX(29);
      AWAIT(65/1),NODE5R/1,BLOCK,1;
      ASSIGN,ATRIB(2)=ATRIB(5),
      ATRIB(5)=5;
      EVENT,6,1;
      ASSIGN,ATRIB(2)=XX(11),1;
RL5   GOON,2;
      ACT,,,RTX5;
      ACT;
      GOON,1;
      ACT,XX(45),.01,DMP5;            TRANSMISSION DELAY WITH ERROR
      ACT,XX(45);
      EVENT,8,1;
      ACT,,ATRIB(2) .EQ. 2 ,NDE2;
      ACT,,ATRIB(2) .EQ. 3 ,NDE3;
      ACT,,ATRIB(2) .EQ. 4 ,NDE4;
      ACT,,ATRIB(2) .EQ. 0 ,RECD;
      ACT,,,DMP5;
RTX5  GOON;
      ACT,XX(65);
      EVENT,9,1;
      ACT,,XX(5) .EQ. 0,DMP5;
      ACT,,,RL5;
;
;
RECD  ASSIGN,ATRIB(5)=9,1;
;                           IF MSG COMPLETED DESTROY PACKET
      EVENT,4,1;
      ACT,,XX(5) .EQ. 0,DMP5;
      ACT;
;            IF BUFFER RESERVED FOR MSG JUMP TO SORTING PACKETS
      EVENT,3,1;
      ACT,,XX(5) .NE. 0,SPD;
;            IF NEITHER OF THE ABOVE TRY TO RESERVE BUFFER FOR MSG
```

```
     ACT;
;
RSVD GOON,1;
     ACT,,XX(99) .LT. 1 ,DPP5;
     ACT,,XX(99) .GE. 1;
     ASSIGN,XX(99)=XX(99)-1;
     COLCT,XX(99),D RESERVATIONS;,1;
     EVENT,1,1;
;
SPD  GOON,1;
     ACT,,ATRIB(8) .EQ. 1,SD1;
     ACT,,ATRIB(8) .EQ. 2,SD2;
     ACT,,ATRIB(8) .EQ. 3,SD3;
;
;  CHECK TO SEE IF PACKET ALREADY HERE FOR THIS MSG  IF YES DESTROY
SD1  EVENT,5,1;
     ACT,,XX(5) .EQ. 0,QD1;
     ACT,,,DMP5;
SD2  EVENT,5,1;
     ACT,,XX(5) .EQ. 0,QD2;
     ACT,,,DMP5;
SD3  EVENT,5,1;
     ACT,,XX(5) .EQ. 0,QD3;
     ACT,,,DMP5;
;
DPP5 ASSIGN,XX(94)=XX(94)+1,1;
     COLCT,XX(94),NO RESERV AT D;,1;
     TERM;



;
QD1  QUEUE(53),,,,MTHD;
QD2  QUEUE(54),,,,MTHD;
QD3  QUEUE(55),,,,MTHD;
;
MTHD MATCH,6,QD1/ND1,QD2/ND1,QD3/ND1;
;
ND1  ACCUM,3,3,HIGH(9);
;    ACT,.003;
     ASSIGN,XX(99)=XX(99)+1;
     EVENT,2,1;
     ACT,,ATRIB(3) .NE. 1,RD;
     ACT;
ND2  COLCT,INT(1),ATOD ROUTING TIME;15/0/.01,1;
     ASSIGN,XX(21)=TNOW-ATRIB(1),1;
RD   GOON,1;
     ACT,XX(24);
     EVENT,7,1;
.    ACT,,ATRIB(3) .EQ. 2,FRBD;
     ACT,,ATRIB(3) .EQ. 3,FRCD;
     ACT;
DDDD FREE,WINDOWA/3,1;
     TERM;
FRBD FREE,WINDOWB/3,1;
     TERM;
FRCD FREE,WINDOWC/3,1;
```

```
     TERM;
;------------------------------------------------------------
;
;SET UP NODE 3, A SUBNET INTERIOR TYPE NODE
;
NDE3  QUEUE(32),,10,BALK(DMP3);
      ACT,XX(27);
      AWAIT(63/1),NODE3R/1,BLOCK,1;
      ASSIGN,ATRIB(2)=ATRIB(5),
      ATRIB(5)=3;
      EVENT,6,1;
      ASSIGN,ATRIB(2)=XX(11),1;
RL3   GOON,2;
      ACT,,,RTX3;
      ACT;
      GOON,1;
      ACT,XX(43),.01,DMP1;           TRANSMISSION.DELAY WITH ERROR
      ACT,XX(43);
      EVENT,8,1;
      ACT,,ATRIB(2) .EQ. 1,NDE1;
      ACT,,ATRIB(2) .EQ. 2,NDE2;
      ACT,,ATRIB(2) .EQ. 4,NDE4;
      ACT,,ATRIB(2) .EQ. 5,NDE5;
RTX3  GOON;
      ACT,XX(63);
      EVENT,9,1;
      ACT,,XX(5) .EQ. 0,DMP3;
      ACT,,,RL3;
;
DMP3  TERM;
;
      ENDNETWORK;
INITIALIZE,0,500;
FIN;
```

F FORTRAN CODE

```
      PROGRAM MYMAIN
      DIMENSION NSET(250000)
      COMMON/SCOM1/ATRIB(100), DD(100), DDL(100), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100),
     2SSL(100),TNEXT, TNOW, XX(100)
      COMMON QSET(250000)
      EQUIVALENCE (NSET(1),QSET(1))
      NNSET = 250000
      NCRDR=5
      NPRNT=6
      NTAPE=7
      CALL SLAM
      STOP
```

```
      END



      SUBROUTINE EVENT(J)


      DIMENSION A(20)
      COMMON/SCOM1/ATRIB(100), DD(100), DDL(100), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100),
     2SSL(100),TNEXT, TNOW, XX(100)

      GO TO (10,20,30,40,50,60,70,80,90),J

C***********************************************************************
C
C                   EVENT 1
C
C     THIS EVENT OCCURS UPON THE RESERVATION OF BUFFER SPACE AT
C     A RECEIVING NODE.  THE FIRST ACTION IS TO WRITE THE ATTRIBUTES
C     OF THE PACKET MAKING THE RESERVATION TO A CMS FILE.  THE SECOND
C     ACTION IS TO PUT THE ATTRIBUTES IN A SLAM FILE WHICH WILL CONTAIN
C     A LIST OF ALL MESSAGES WHICH HAVE HAD SPACE PREVIOUSLY RESERVED
C     SO AS TO PRECLUDE DUPLICATE RESERVATIONS FOR ANY ONE MESSAGE.
C
10    J=ATRIB(4)*10
      CALL FFILE(J,ATRIB)
      RETURN

C***********************************************************************
C
C                   EVENT 2
C
C
C     REMOVES MESSAGES FROM 2 FILES AS AN INDICATOR OF
C     COMPLETED MESSAGES. (FOR EXAMPLE: IF THE MESSAGE SERIAL NUMBER
C     IS NOT IN THE FILE (NOT WITHIN THE WINDOW OF TRANSMITTABLE
C     MESSAGES) THEN IT HAS ALREADY BEEN RECEIVED IN ENTIRETY OR
C     IT'S SERIAL NUMBER IS IN ERROR). FILES USED IN TESTING AT
C     RECEIVING SIDE OF HOST AND DURING HOST-HOST RETX LOOP.
C
20    J=ATRIB(4)+4
      P=ATRIB(6)
      K=1
21    CONTINUE
        K=NFIND(K,J,6,0,P,0.0)
        IF (K .EQ. 0) GO TO 22
        CALL RMOVE(K,J,A)
        IF(K .GT. NNQ(J)) GO TO 22
        GO TO 21
22    CONTINUE
      J=ATRIB(4)*10
```

```
      P=ATRIB(6)
      K=1
25    CONTINUE
        K=NFIND(K,J,6,0,P,0.0)
        IF (K .EQ. 0) GO TO 26
        CALL RMOVE(K,J,A)
        IF(K .GT. NNQ(J)) GO TO 26
        GO TO 25
26    CONTINUE
C
C     IF THIS IS A COMPLETION (ATRIB(5)=9)
C     AT HOST D (ATRIB(4)=4), A TO D MESSAGE,  PUT STATS
C     (BATCH STATS WITH BATCH = 100) IN A TABLE IN CMS FILE 15
C
       IF (  (ATRIB(5) .EQ. 9) .AND. (ATRIB(4) .EQ. 4)
     &   .AND. (ATRIB(3) .EQ. 1) )THEN
C     ***** IF INDIVIDUAL ROUTE TIMES ARE NEEDED UNCOMMENT THIS WRITE:
C      WRITE (14,29)TNOW,TNOW-ATRIB(1),ATRIB(6),ATRIB(9),XX(99),XX(94)
         RTIM=TNOW-ATRIB(1)
         XX(91)=XX(91)+1
         BNUM=XX(91)
         BMEAN=XX(92)
         BVAR=XX(93)
         IF( NINT(BNUM) .EQ. 1 ) THEN
             BMEAN=RTIM
             BVAR=0.0
         ELSE
             BVAR=((BNUM-2.0)*BVAR+((BNUM-1.0)/BNUM)*(RTIM-BMEAN)**2)/
     &              (BNUM-1.0)
             BMEAN=(RTIM+(BNUM-1.0)*BMEAN)/BNUM
         END IF
         IF ( NINT(BNUM) .GE. 100 ) THEN
             WRITE (15,29) TNOW,BMEAN,BVAR,BNUM
             XX(91)=0
             XX(92)=0
             XX(93)=0
         ELSE
             XX(92)=BMEAN
             XX(93)=BVAR
         END IF
      END IF
29    FORMAT (F10.4,1X,F10.5,1X,F11.5,1X,F6.1,1X,F10.5,1X,F10.5)
      RETURN

C*************************************************************************
C
C                    EVENT 3
C
C
C      THIS EVENT CHECKS THE APPROPRIATE FILE TO DETERMINE IF A
C      MESSAGE HAS ALREADY HAD SPACE RESERVED FOR IT IN THE
C      RECEIVING HOST'S BUFFER.  RETURNS XX(5) AS AN INDICATOR OF
C      A RESERVATION HAVING BEEN MADE:  XX(5)=0 INDICATES NO
C      RESERVATION FOR THIS MESSAGE NUMBER.
C
30    J=ATRIB(4)*10
```

```
      P=ATRIB(6)
      XX(5)=NFIND(1,J,6,0,P,0.0)
      RETURN

C*********************************************************************
C
C                         EVENT 4
C
C
C        THIS SUBROUTINE TESTS TO DETERMINE IF A MESSAGE HAS BEEN
C     COMPLETELY RECEIVED BY THE RECEIVING HOST.  IT IS CALLED FROM TWO
C     DISTINCTLY DIFFERENT PLACES.
C        THE FIRST BEING AT THE SENDING HOST ( ATTRIBUTE(5) = 0 )
C     WHERE IT IS USED TO DETERMINE IF A MESSAGE SHOULD BE RETRANSMITTED.
C     PRESENCE IN THE FILE (XX(5) NOT EQUAL 0) MEANS KEEP RETRANSMITTING.
C        THE SECOND IS AT THE RECEIVING HOST ( ATTRIBUTE(5) NOT EQUAL 0 )
C     WHERE IT IS USED TO PREVENT ALREADY RECEIVED MESSAGES FROM ENTERING.
C        EVENT RETURNS THE GLOBAL XX(5) AS AN INDICATOR OF WHETHER THAT
C     MESSAGE IS IN THE APPROPRIATE FILE.
C        TWO DIFFERENT FILES ARE USED FOR ESSENTIALLY THE SAME PURPOSE
C     BECAUSE THE RECEIVING HOST KNOWS INSTANTANEOUSLY IF IT HAS
C     RECEIVED A MESSAGE.  WHEREAS THE SENDING HOST HAS A TIME DELAY
C     BEFORE THE END TO END ACKNOWLEDGEMENT IS RECEIVED AND THE SENDER
C     BECOMES AWARE THAT THE MESSAGE HAS BEEN RECEIVED.

40    J=ATRIB(4)+4
      IF (ATRIB(5) .EQ. 0) J=(ATRIB(3)*10)+6
      P=ATRIB(6)
      XX(5)=NFIND(1,J,6,0,P,0.001)
      RETURN

C*********************************************************************
C
C                      EVENT 5
C
C
C        THIS EVENT CHECKS THE APPROPRIATE FILE TO DETERMINE IF A
C        A PARTICULAR PACKET OF A MESSAGE HAS ALREADY BEEN
C        RECEIVED BY THE RECEIVING HOST. THIS PREVENTS DUPLICATE
C        PACKETS OF THE SAME MESSAGE FROM OCCUPYING BUFFER SPACE.
C        RETURNS XX(5) AS AN INDICATOR OF THE PACKET BEING IN THE
C        RECEIVER'S BUFFER:  XX(5)=0 INDICATES THIS PACKET NOT IN
C        THE RECEIVER'S BUFFER.
C
50    J=ATRIB(4)
      IF (J .LT. 2.5) JJ=(J*10)+ATRIB(8)+2
      IF (J .GT. 2.5) JJ=((J+1)*10)+ATRIB(8)+2
      P=ATRIB(6)
      XX(5)=NFIND(1,JJ,6,0,P,0.001)
      RETURN

C*********************************************************************

60    CALL ROUTE
      RETURN
70    CALL HRETX
```

```
      RETURN
80    CALL DRETX
      RETURN
90    CALL QRETX
      RETURN
      END

C-------------------------------------------------------------------
      SUBROUTINE ROUTE
C
C       AN ADAPTIVE ROUTING SCHEME WHICH USE THE CURRENT NODE
C     LOCATION AND DESTINATION OF THIS PACKET TO ENTER AN ARRAY
C     WHICH CONTAINS THE NEXT NODE TO ROUTE THIS PACKET TOWARD.
C       THE ARRAY IS INITIALIZED AND FIXED FOR THE FIRST 10
C     SECONDS OF THE SIMULATION RUN.  THEREAFTER IT IS UPDATED
C     UPON THE FIRST CALL AFTER TEN SECONDS SIMULATION TIME HAS
C     ELAPSED.  THE ARRAY IS UPDATED BY NOTING THE AVERAGE WAIT
C     IN EACH POSSIBLE CHOICE OF FOLLOW ON QUEUES AND SETTING THE
C     ARRAY TO THE ONE WITH THE SHORTEST WAIT OR SHORTEST PATH IF
C     THERE IS A TIE.
C

      INTEGER DELN,JFILE
      REAL AVGWT(5)


      COMMON/SCOM1/ATRIB(100), DD(100), DDL(100), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100),
     2SSL(100),TNEXT, TNOW, XX(100)

      J=ATRIB(5)*10+7
      CALL FFILE(J,ATRIB)

      IIIY=ATRIB(5)
      JJJY=ATRIB(4)
      IF (TNOW .LE. 10.0) XX(10)=1
      IF (TNOW .GT. 10*XX(10)) THEN
         DO 10 I=1,5
            AVGWT(I)=0.0
            JJ=10*I+2
            IF ( FFAWT(JJ) .LT. .00001) GO TO 10
            NNOW=NINT(FFAVG(JJ)*TNOW/FFAWT(JJ))
            WAITN=NNOW*FFAWT(JJ)
            DELN=NNOW-NINT(XX(79+I))
            IF ( DELN .LT. .00001) GO TO 10
            DELW=WAITN-XX(84+I)
            AVGWT(I)=DELW/DELN
            XX(79+I)=NNOW
            XX(84+I)=WAITN
10       CONTINUE
         MINV=3
         DO 20 II=1,5,2
            IF( AVGWT(II) .LT. AVGWT(MINV) ) MINV=II
20       CONTINUE
         MINH=3
         DO 30 III=2,4
```

```
          IF( AVGWT(III) .LT. AVGWT(MINH) ) MINH=III
30      CONTINUE
        PV=REAL(MINV)
        IF (GETARY(2,3) .NE. PV) THEN
          CALL PUTARY(2,3,PV)
          CALL PUTARY(4,2,PV)
        END IF
        PH=REAL(MINH)
        IF (GETARY(1,4) .NE. PH) THEN
          CALL PUTARY(1,4,PH)
          CALL PUTARY(5,1,PH)
        END IF
        XX(10)=XX(10)+1
      END IF
      XX(11)=GETARY(IIIY,JJJY)
      RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE HRETX
C
C                         EVENT 7
C
C        THIS SUBROUTINE PUTS ATTRIBUTES OF MESSAGES THAT HAVE PROCEEDED
C     THROUGH THE SENDING HOST'S SLIDING WINDOW INTO A FILE (16,26,36,OR
C     46) WHICH HOLDS "MESSAGE SERIAL NUMBERS OKAY FOR RETRANSMIT".
C        WHEN THE MESSAGE IS RECEIVED AT THE RECEIVING HOST THE
C     ATTRIBUTES ARE REMOVED FROM THE FILE THUS PREVENTING FURTHER
C     RETRANSMISSIONS.
C
C
      DIMENSION A(20)

      COMMON/SCOM1/ATRIB(100), DD(100), DDL(100), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100),
     2SSL(100),TNEXT, TNOW, XX(100)

      J=ATRIB(3)*10+6
      K=ATRIB(4)+4
C
C IF CALLED FROM SENDING HOST THEN PUT ATTRIBUTES IN APPROPRIATE FILE
C
      IF (ATRIB(5) .EQ. 0) THEN
          CALL FFILE(J,ATRIB)
          CALL FFILE(K,ATRIB)
C
C ELSE IT'S A CALL FROM A RECEIVING HOST SO
C FIND THE RANK(S) OF PACKETS WITH SAME MESSAGE SERIAL NUMBER
C AND REMOVE THEM FROM THE FILE.
C
      ELSE
          P=ATRIB(6)
          K=1
10        CONTINUE
            K=NFIND(K,J,6,0,P,0.0)
            IF (K .EQ. 0) GO TO 20
            CALL RMOVE(K,J,A)
```

```
            IF(K .GT. NNQ(J)) GO TO 20
         GO TO 10
      END IF
20    CONTINUE
      RETURN
      END
C------------------------------------------------------------
      SUBROUTINE DRETX
C
C                        EVENT 8
C
C
C        SUBROUTINE REMOVES PACKETS FROM THE NODE TO NODE
C     RETRANSMISSION LIST AND FREES THE NODE TO NODE LINK RESOURCE
C     BUT ONLY IF THERE IS BUFFER SPACE AVAILABLE AT THE NEXT NODE IN
C     THE ROUTE.  OTHERWISE IT JUST LETS THE PACKET FLOW WHICH WILL
C     RESULT IN IT BEING LOST WHEN IT FINDS THE BUFFER FULL.
C
      DIMENSION A(20)

      COMMON/SCOM1/ATRIB(100), DD(100), DDL(100), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100),
     2SSL(100),TNEXT, TNOW, XX(100)

      NEXT=NINT(ATRIB(2))
      NNXT=NEXT*10+2
      IF( (NEXT .EQ. 0) .OR.
     &    ((NEXT .GT. 0) .AND. (NNQ(NNXT) .LT. 10) ) )THEN
C .......  IT'S AT IT'S DESTINATION INTERFACE OR
C ....... IF THE QUEUE AT NEXT NODE IS NOT FULL THEN
C .......   REMOVE FROM PREVIOUS NODE'S OKAY TO RETX FILE.......
        J=ATRIB(5)*10+7
        JJ=ATRIB(5)
        IF (ATRIB(5) .GT. 5) WRITE (14,*) 'EVENT 8 ATRIB(5)=',ATRIB(5)

        P=ATRIB(6)
        Q=ATRIB(8)
        K=1
10      CONTINUE
           K=NFIND(K,J,6,0,P,0.0)
           IF (K .EQ. 0) GO TO 20
           CALL COPY (K,J,A)
           IF (Q .EQ. A(8)) THEN
              CALL RMOVE(K,J,A)
              IF (K .GT. NNQ(J)) GO  TO 20
              GO TO 10
           END IF
           IF (K .GE. NNQ(J)) GO  TO 20
           K=K+1
        GO TO 10
20      CONTINUE
C   ........AND THEN FREE A NODE TO NODE LINK AT THE PREVIOUS NODE
        CALL FREE(JJ,1)
      END IF
      RETURN
```

```
      END
C--------------------------------------------------------------------------
      SUBROUTINE QRETX
C
C                          EVENT 9
C
C
C        SUBROUTINE CALLED TO DETERMINE IF A PACKET SHOULD BE
C     RETRANSMITTED NODE TO NODE UPON COMPLETION OF THE TIMEOUT.
C     WILL RETRANSMIT IF PACKET STILL IN THE "OKAY TO RETRANSMIT
C     FILE" OF THIS CALLING NODE.  RETURNS XX(5) = 1 AS AN
C     INDICATOR OF OKAY TO RETRANSMIT.
C
      DIMENSION A(20)

      COMMON/SCOM1/ATRIB(100), DD(100), DDL(100), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100),
     2SSL(100),TNEXT, TNOW, XX(100)
      XX(5)=0
      J=ATRIB(5)*10+7
      P=ATRIB(6)
      Q=ATRIB(8)
      K=1
10    CONTINUE
         K=NFIND(K,J,6,0,P,0.0)
         IF (K .EQ. 0) GO TO 40
         CALL COPY(K,J,A)
         IF (Q .EQ. A(8)) THEN
             XX(5)=1
             GO TO 40
         END IF
         K=K+1
         IF (K .GT. NNQ(J)) GO  TO 40
      GO TO 10
40    CONTINUE
      RETURN
      END
```

APPENDIX B. AUTOCORRELATION FUNCTION ANALYSIS OF RUNS

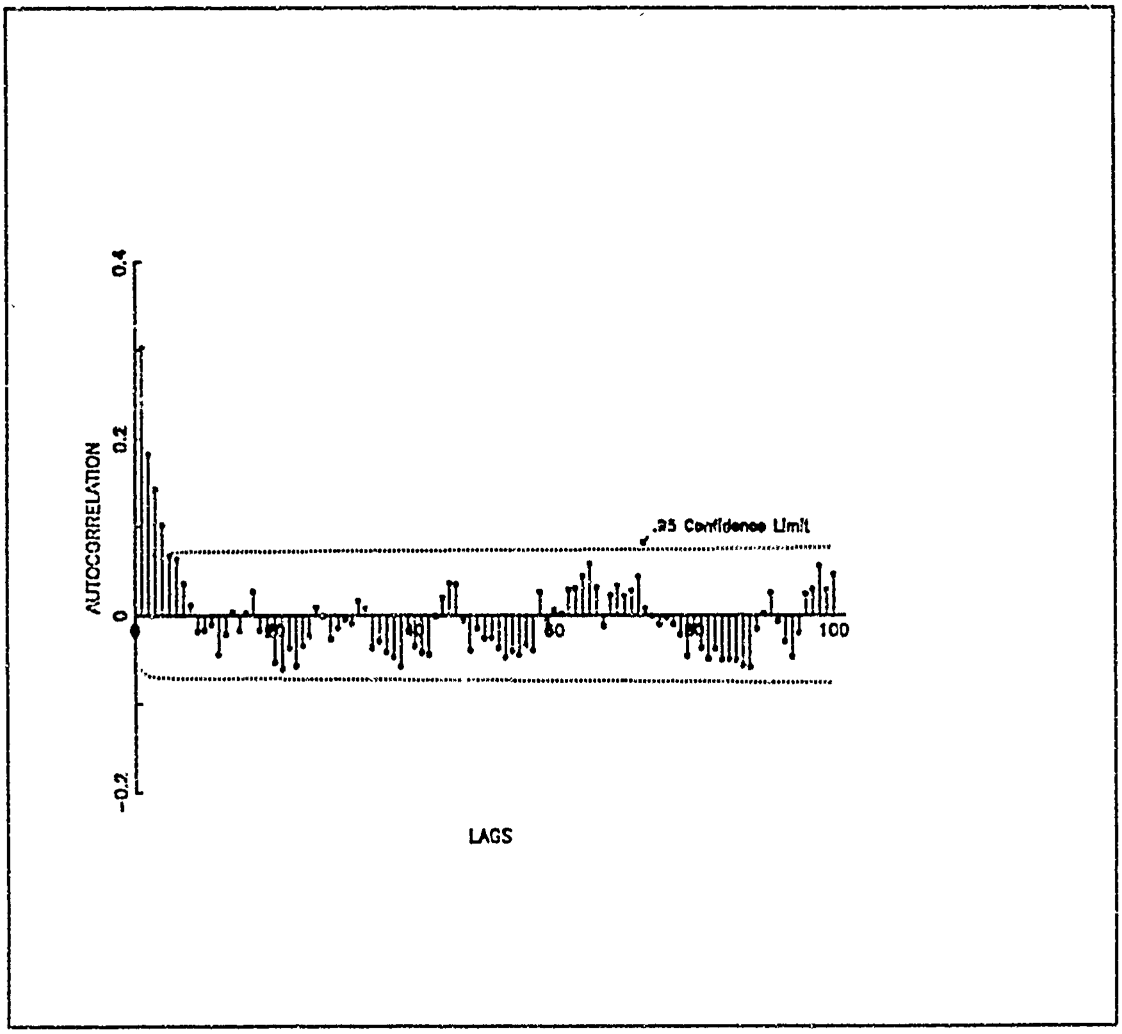

Figure 16. Autocorrelation Function Analysis of an Independent Simulation Run

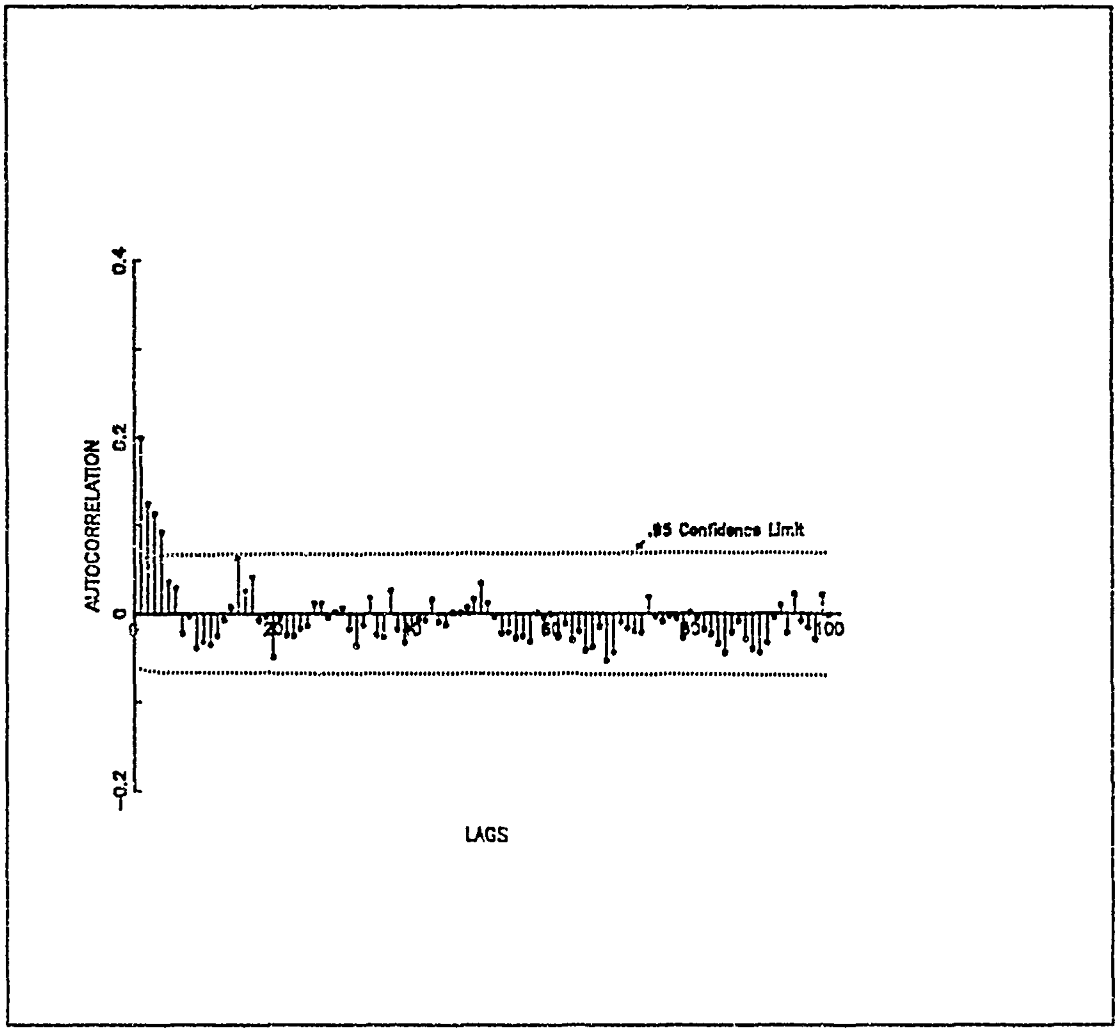

Figure 17. Autocorrelation Function Analysis of an Independent Simulation Run

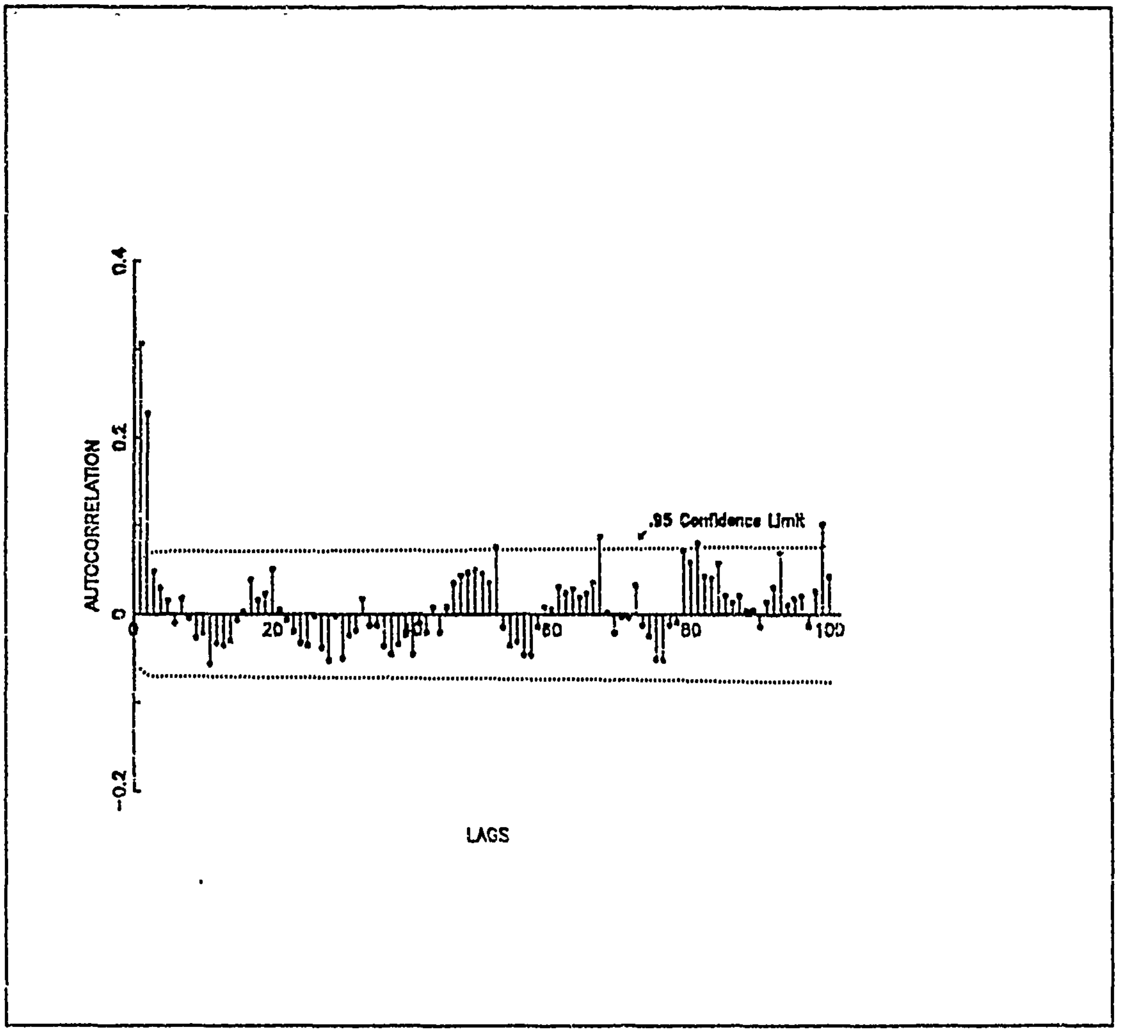

Figure 18. Autocorrelation Function Analysis of an Independent Simulation Run

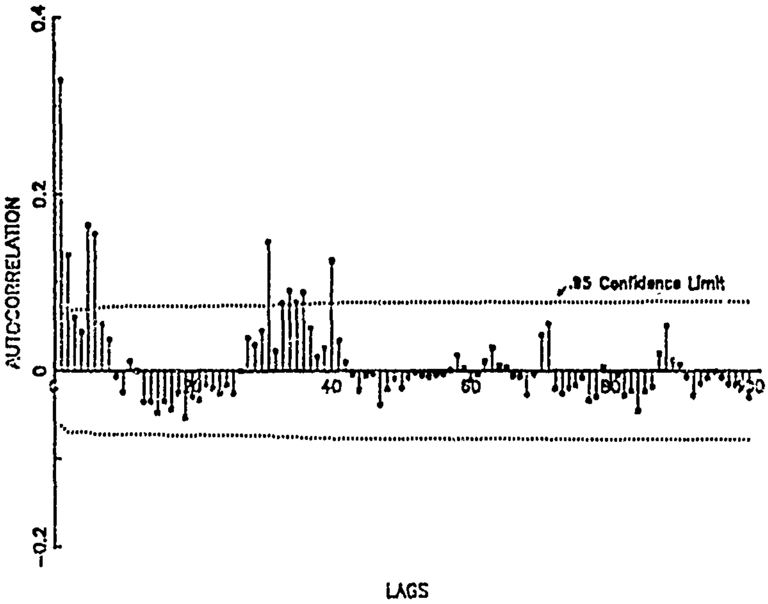

Figure 19. Autocorrelation Function Analysis of an Independent Simulation Run

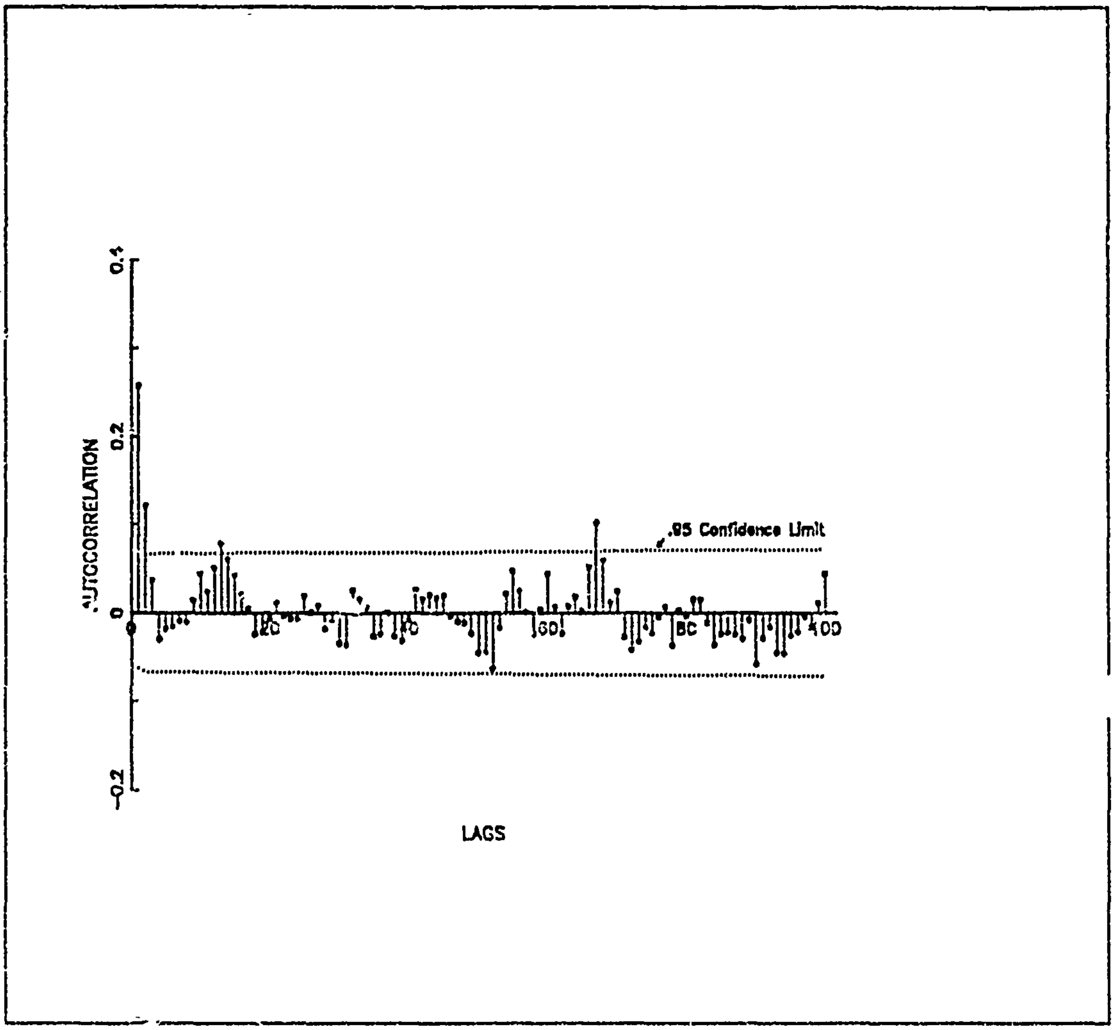

Figure 20. Autocorrelation Function Analysis of an Independent Simulation Run

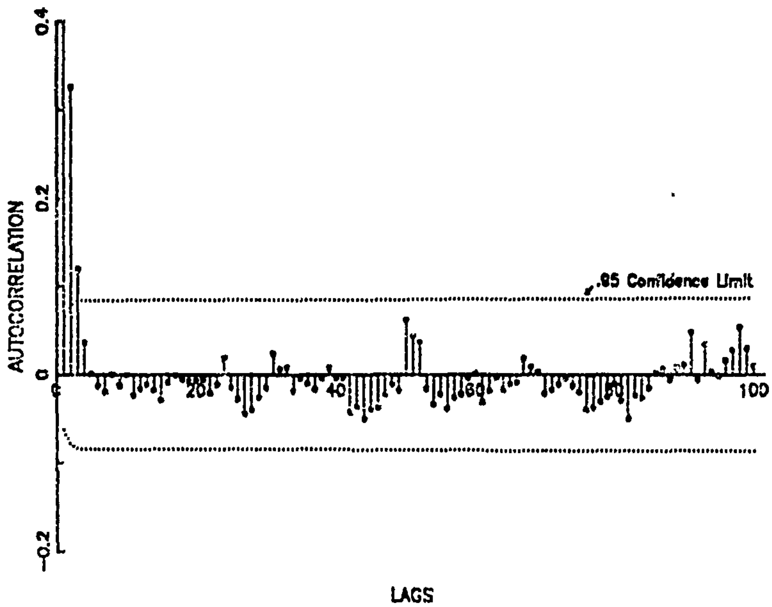

Figure 21. Autocorrelation Function Analysis of an Independent Simulation Run

APPENDIX C. DETAILED REPORT OF EXPERIMENTAL DATA

A. NODE TIMEOUT EXPERIMENT 1 DATA

NODE TIMEOUT INTERVAL = 0.0012 SECONDS

TIME OF OBSERVATION	BATCH MEAN	BATCH VARIANCE	BATCH SIZE
25.1189	0.10302	0.00174	100.0
56.6924	0.09041	0.00070	100.0
93.3726	0.09695	0.00111	100.0
121.6888	0.08687	0.00063	100.0
147.5486	0.09902	0.00227	100.0
169.1679	0.10347	0.00125	100.0
196.0497	0.09352	0.00109	100.0
225.5437	0.09108	0.00067	100.0
248.1594	0.10572	0.00196	100.0
275.6798	0.09980	0.00189	100.0
301.4587	0.11101	0.00613	100.0
325.9315	0.08733	0.00096	100.0
350.7789	0.09785	0.00182	100.0
375.2984	0.09617	0.00152	100.0
403.1508	0.09287	0.00106	100.0
436.0787	0.09023	0.00082	100.0
463.4310	0.09422	0.00185	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09604
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08326, 0.10882

NODE TIMEOUT INTERVAL = 0.0013 SECONDS

TIME OF OBSERVATION	BATCH MEAN	BATCH VARIANCE	BATCH SIZE
25.1189	0.10303	0.00174	100.0
56.6924	0.09042	0.00070	100.0
93.3726	0.09696	0.00111	100.0
121.6888	0.08696	0.00063	100.0
147.5486	0.09879	0.00217	100.0
169.1679	0.10382	0.00135	100.0
196.0497	0.09365	0.00109	100.0
225.5437	0.09108	0.00067	100.0
248.1594	0.10569	0.00196	100.0
275.6798	0.09980	0.00188	100.0
301.4587	0.10694	0.00319	100.0
325.9315	0.08727	0.00096	100.0
350.7789	0.09704	0.00178	100.0
375.2984	0.09597	0.00150	100.0
403.1508	0.09290	0.00106	100.0
436.0787	0.09031	0.00082	100.0
463.4310	0.09352	0.00159	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09575
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08388, 0.10761

NODE TIMEOUT INTERVAL = 0.0014 SECONDS

25.1189	0.10316	0.00173	100.0
56.6924	0.09042	0.00070	100.0
93.3726	0.09697	0.00111	100.0
121.6888	0.08698	0.00063	100.0
147.5486	0.09882	0.00218	100.0
169.1679	0.10389	0.00135	100.0
196.0497	0.09362	0.00109	100.0
225.5437	0.09112	0.00067	100.0
248.1594	0.10560	0.00197	100.0
275.6798	0.09982	0.00188	100.0
301.4587	0.10641	0.00388	100.0
325.9329	0.08727	0.00095	100.0
350.7789	0.09724	0.00178	100.0
375.2984	0.09576	0.00150	100.0
403.1508	0.09291	0.00106	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09573
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08395, 0.10750

NODE TIMEOUT INTERVAL = 0.0016 SECONDS

25.1189	0.10317	0.00173	100.0
56.6924	0.09043	0.00070	100.0
93.3726	0.09699	0.00111	100.0
121.6888	0.08702	0.00063	100.0
147.5486	0.09893	0.00222	100.0
169.1679	0.10367	0.00125	100.0
196.0497	0.09355	0.00109	100.0
225.5437	0.09138	0.00068	100.0
248.1594	0.10568	0.00196	100.0
275.6798	0.09982	0.00188	100.0
301.4587	0.11128	0.00888	100.0
325.9315	0.08691	0.00092	100.0
350.7789	0.09712	0.00178	100.0
375.2984	0.09577	0.00150	100.0
403.1508	0.09291	0.00106	100.0
436.0787	0.09021	0.00083	100.0
463.4310	0.09293	0.00146	100.0
436.0787	0.09025	0.00082	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09596
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08305, 0.10886

NODE TIMEOUT INTERVAL = 0.0017 SECONDS

25.1189	0.10317	0.00173	100.0
56.6924	0.09043	0.00070	100.0
93.3726	0.09700	0.00111	100.0
121.6888	0.08704	0.00063	100.0

147.5486	0.09904	0.00227	100.0
169.1679	0.10381	0.00132	100.0
196.0497	0.09357	0.00109	100.0
225.5437	0.09160	0.00069	100.0
248.1594	0.10567	0.00196	100.0
275.6798	0.09995	0.00189	100.0
301.4587	0.10572	0.00285	100.0
325.9315	0.08726	0.00095	100.0
350.7789	0.09704	0.00178	100.0
375.2984	0.09617	0.00150	100.0
403.1508	0.09299	0.00105	100.0
436.0787	0.09024	0.00082	100.0
463.4327	0.09266	0.00139	100.0
463.4310	0.09307	0.00149	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09571
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08405, 0.10736

NODE TIMEOUT INTERVAL = 0.0018 SECONDS

25.1189	0.10318	0.00173	100.0
56.6924	0.09043	0.00070	100.0
93.3726	0.09713	0.00111	100.0
121.6888	0.08706	0.00063	100.0
147.5486	0.09882	0.00217	100.0
169.1679	0.10376	0.00131	100.0
196.0497	0.09356	0.00110	100.0
225.5437	0.09115	0.00068	100.0
248.1594	0.10576	0.00197	100.0
275.6816	0.09982	0.00188	100.0
301.4587	0.11188	0.00900	100.0
325.9315	0.08728	0.00095	100.0
350.7789	0.09707	0.00178	100.0
375.2984	0.09583	0.00150	100.0
403.1508	0.09292	0.00106	100.0
436.0787	0.09028	0.00082	100.0
463.4310	0.09437	0.00187	100.0
463.4310	0.09361	0.00163	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09609
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08309, 0.10908

NODE TIMEOUT INTERVAL = 0.0019 SECONDS

25.1189	0.10319	0.00173	100.0
56.6924	0.09043	0.00070	100.0
93.3726	0.09795	0.00119	100.0
121.6888	0.08653	0.00067	100.0
147.5486	0.09869	0.00213	100.0
169.1679	0.10363	0.00132	100.0
196.0497	0.09379	0.00113	100.0
225.5437	0.09088	0.00065	100.0
248.1594	0.10590	0.00196	100.0
275.6798	0.09985	0.00188	100.0

301.4587	0.11793	0.00731	100.0
325.9315	0.08731	0.00096	100.0
350.7789	0.09702	0.00178	100.0
375.2984	0.09588	0.00149	100.0
403.1508	0.09289	0.00106	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09636
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08147, 0.11124

NODE TIMEOUT INTERVAL = 0.002 SECONDS

25.1189	0.10448	0.00176	100.0
56.6924	0.09092	0.00071	100.0
93.3726	0.09882	0.00120	100.0
121.6888	0.08896	0.00079	100.0
147.5606	0.09924	0.00216	100.0
169.1679	0.10557	0.00149	100.0
196.0517	0.09331	0.00097	100.0
225.5437	0.09110	0.00062	100.0
248.1594	0.10596	0.00193	100.0
275.6798	0.10042	0.00190	100.0
301.4587	0.11584	0.00770	100.0
325.9315	0.08719	0.00095	100.0
350.7789	0.09856	0.00175	100.0
375.2984	0.09653	0.00148	100.0
403.1508	0.09378	0.00110	100.0
436.1027	0.09247	0.00083	100.0
463.4310	0.09453	0.00166	100.0
490.1384	0.09048	0.00067	100.0
436.0787	0.09016	0.00083	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09620
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08240, 0.11000

NODE TIMEOUT INTERVAL = 0.0022 SECONDS

25.1189	0.10320	0.00173	100.0
56.6924	0.09044	0.00070	100.0
93.3726	0.09798	0.00119	100.0
121.6888	0.08658	0.00067	100.0
147.5486	0.09895	0.00222	100.0
169.1679	0.10331	0.00126	100.0
196.0519	0.09372	0.00109	100.0
225.5437	0.09121	0.00068	100.0
248.1594	0.10603	0.00197	100.0
275.6798	0.09984	0.00189	100.0
301.4587	0.10945	0.00424	100.0
325.9315	0.08733	0.00095	100.0
350.7789	0.09709	0.00178	100.0
375.2984	0.09610	0.00149	100.0
403.1508	0.09298	0.00106	100.0
436.0787	0.09045	0.00083	100.0
463.4310	0.09366	0.00162	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09600
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08357, 0.10843

NODE TIMOUT INTERVAL = 0.0024 SECONDS

25.1189	0.10322	0.00173	100.0
56.6924	0.09045	0.00070	100.0
93.3726	0.09800	0.00119	100.0
121.6888	0.08662	0.00067	100.0
147.5486	0.09945	0.00243	100.0
169.1679	0.10374	0.00133	100.0
196.0497	0.09376	0.00110	100.0
225.5437	0.09130	0.00068	100.0
248.1594	0.10589	0.00197	100.0
275.6798	0.09995	0.00189	100.0
301.4587	0.11572	0.01117	100.0
325.9315	0.08749	0.00095	100.0
350.7789	0.09710	0.00178	100.0
375.2984	0.09577	0.00151	100.0
403.1508	0.09316	0.00106	100.0
436.0787	0.09032	0.00082	100.0
463.4310	0.09341	0.00155	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09637
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08222, 0.11052

NODE TIMEOUT INTERVAL = 0.0026 SECONDS

25.1189	0.10323	0.00173	100.0
56.6924	0.09046	0.00070	100.0
93.3726	0.09802	0.00120	100.0
121.6888	0.08666	0.00067	100.0
147.5486	0.09883	0.00217	100.0
169.1705	0.10381	0.00120	100.0
196.0497	0.09400	0.00110	100.0
225.5437	0.09122	0.00067	100.0
248.1594	0.10570	0.00196	100.0
275.6798	0.09997	0.00188	100.0
301.4587	0.11556	0.00581	100.0
325.9315	0.08736	0.00096	100.0
350.7789	0.09717	0.00178	100.0
375.3010	0.09572	0.00150	100.0
403.1508	0.09313	0.00108	100.0
436.0787	0.09028	0.00082	100.0
463.4310	0.09366	0.00165	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09635
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08230, 0.11040

NODE TIMEOUT INTRVAL = 0.0028 SECONDS

25.1189	0.10324	0.00173	100.0
56.6924	0.09046	0.00070	100.0

93.3726	0.09804	0.00120	100.0
121.6888	0.08670	0.00067	100.0
147.5486	0.09860	0.00209	100.0
169.1679	0.10347	0.00126	100.0
196.0497	0.09389	0.00108	100.0
225.5437	0.09121	0.00067	100.0
248.1594	0.10591	0.00199	100.0
275.6798	0.10012	0.00187	100.0
301.4587	0.11497	0.00847	100.0
325.9315	0.08721	0.00096	100.0
350.7789	0.09740	0.00179	100.0
375.2984	0.09620	0.00152	100.0
403.1508	0.09311	0.00107	100.0
436.0787	0.09057	0.00084	100.0
463.4310	0.09389	0.00164	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09637
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08252, 0.11022

NODE TIMEOUT INTERVAL = 0.003 SECONDS

25.1189	0.10325	0.00173	100.0
56.6924	0.09047	0.00070	100.0
93.3726	0.09806	0.00119	100.0
121.6888	0.08674	0.00068	100.0
147.5516	0.09864	0.00208	100.0
169.1679	0.10504	0.00179	100.0
196.0497	0.09356	0.00109	100.0
225.5437	0.09118	0.00067	100.0
248.1594	0.10574	0.00196	100.0
275.6798	0.09997	0.00188	100.0
301.4587	0.10790	0.00392	100.0
325.9345	0.08747	0.00095	100.0
350.7789	0.09734	0.00179	100.0
375.3014	0.09596	0.00149	100.0
403.1508	0.09286	0.00106	100.0
436.0787	0.09035	0.00082	100.0
463.4310	0.09337	0.00149	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09597
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08367, 0.10827

NODE TIMEOUT INTERVAL = 0.0032 SECONDS

25.1189	0.10326	0.00173	100.0
56.6924	0.09047	0.00070	100.0
93.3726	0.09808	0.00119	100.0
121.6888	0.08677	0.00068	100.0
147.5486	0.09885	0.00217	100.0
169.1679	0.10386	0.00126	100.0
196.0497	0.09360	0.00109	100.0
225.5437	0.09156	0.00068	100.0
248.1594	0.10572	0.00196	100.0
275.6798	0.10047	0.00198	100.0
301.4587	0.11385	0.00672	100.0

325.9315	0.08741	0.00095	100.0
350.7789	0.09756	0.00182	100.0
375.2984	0.09588	0.00150	100.0
403.1508	0.09311	0.00108	100.0
436.0787	0.09033	0.00083	100.0
463.4310	0.09285	0.00142	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09629
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08267, 0.10990

NODE TIMEOUT INTERVAL = 0.0034 SECONDS

25.1189	0.10328	0.00173	100.0
56.6924	0.09048	0.00070	100.0
93.3726	0.09798	0.00120	100.0
121.6888	0.08681	0.00068	100.0
147.5486	0.09910	0.00227	100.0
169.1679	0.10398	0.00126	100.0
196.0497	0.09378	0.00107	100.0
225.5437	0.09135	0.00069	100.0
248.1594	0.10602	0.00201	100.0
275.6798	0.09732	0.00168	100.0
301.4587	0.10697	0.00368	100.0
325.9315	0.08743	0.00095	100.0
350.7789	0.09725	0.00182	100.0
375.2984	0.09562	0.00150	100.0
403.1508	0.09295	0.00105	100.0
436.0787	0.09031	0.00083	100.0
463.4310	0.09321	0.00145	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09573
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08384, 0.10762

NODE TIMEOUT INTERVAL = 0.0036 SECONDS

25.1189	0.10329	0.00173	100.0
56.6924	0.09049	0.00070	100.0
93.3726	0.09799	0.00120	100.0
121.6888	0.08685	0.00068	100.0
147.5486	0.09922	0.00232	100.0
169.1679	0.10395	0.00120	100.0
196.0497	0.09388	0.00109	100.0
225.5437	0.09143	0.00068	100.0
248.1594	0.10592	0.00195	100.0
275.6798	0.10003	0.00187	100.0
301.4587	0.12145	0.00783	100.0
325.9351	0.08727	0.00096	100.0
350.7789	0.09740	0.00179	100.0
375.3020	0.09587	0.00149	100.0
403.1508	0.09325	0.00105	100.0
436.0787	0.09046	0.00083	100.0
463.4310	0.094[illegible]	0.00185	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09684
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08098, 0.11270

NODE TIMEOUT INTERVAL = 0.004 SECONDS

25.1189	0.10331	0.00173	100.0
56.6924	0.09050	0.00070	100.0
93.3726	0.09801	0.00120	100.0
121.6888	0.08693	0.00068	100.0
147.5486	0.09864	0.00208	100.0
169.1679	0.10447	0.00138	100.0
196.0497	0.09371	0.00109	100.0
225.5477	0.09154	0.00068	100.0
248.1594	0.10581	0.00196	100.0
275.6798	0.10024	0.00199	100.0
301.4587	0.11586	0.01118	100.0
325.9315	0.08745	0.00096	100.0
350.7789	0.09694	0.00178	100.0
375.3024	0.09623	0.00150	100.0
403.1508	0.09297	0.00106	100.0
436.0787	0.09098	0.00084	100.0
463.4310	0.09302	0.00138	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09646
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08232, 0.11059

NODE TIMEOUT INTERVAL = 0.005

25.1189	0.10337	0.00173	100.0
56.6924	0.09053	0.00070	100.0
93.3726	0.09818	0.00120	100.0
121.6888	0.08712	0.00069	100.0
147.5536	0.09884	0.00212	100.0
169.1679	0.10475	0.00139	100.0
196.0497	0.09354	0.00110	100.0
225.5437	0.09158	0.00066	100.0
248.1594	0.10600	0.00203	100.0
275.6848	0.09985	0.00187	100.0
301.4587	0.10794	0.00411	100.0
325.9315	0.08711	0.00093	100.0
350.7789	0.09741	0.00182	100.0
375.2984	0.09596	0.00150	100.0
403.1508	0.09323	0.00105	100.0
436.0787	0.09104	0.00086	100.0
463.4310	0.09427	0.00181	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09611
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08391, 0.10831

NODE TIMEOUT INTERVAL = 0.006 SECONDS

25.1189	0.10343	0.00174	100.0
56.6924	0.09056	0.00070	100.0

93.3726	0.09823	0.00120	100.0
121.6888	0.08731	0.00069	100.0
147.5486	0.09870	0.00206	100.0
169.1679	0.10345	0.00115	100.0
196.0497	0.09392	0.00108	100.0
225.5437	0.09178	0.00071	100.0
248.1594	0.10586	0.00194	100.0
275.6798	0.10044	0.00190	100.0
301.4587	0.10647	0.00390	100.0
325.9315	0.08740	0.00096	100.0
350.7789	0.09735	0.00178	100.0
375.2984	0.09650	0.00147	100.0
403.1568	0.09339	0.00105	100.0
436.0787	0.09047	0.00081	100.0
463.4310	0.09379	0.00156	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09604
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08441, 0.10768

NODE TIMEOUT INTERVAL = 0.007 SECONDS

25.1189	0.10349	0.00174	100.0
56.6924	0.09059	0.00071	100.0
93.3726	0.09827	0.00120	100.0
121.6888	0.08750	0.00070	100.0
147.5486	0.09873	0.00208	100.0
169.1679	0.10460	0.00126	100.0
196.0497	0.09382	0.00108	100.0
225.5507	0.09133	0.00068	100.0
248.1594	0.10644	0.00204	100.0
275.6798	0.10020	0.00190	100.0
301.4587	0.11301	0.00701	100.0
325.9385	0.08790	0.00094	100.0
350.7789	0.09751	0.00179	100.0
375.2984	0.09614	0.00149	100.0
403.1508	0.09323	0.00106	100.0
436.0787	0.09071	0.00084	100.0
463.4310	0.09384	0.00143	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09651
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08321, 0.10980

NODE TIMEOUT INTERVAL = 0.008 SECONDS

25.1189	0.10355	0.00174	100.0
56.6924	0.09062	0.00071	100.0
93.3726	0.09832	0.00120	100.0
121.6888	0.08769	0.00070	100.0
147.5486	0.09877	0.00208	100.0
169.1679	0.10399	0.00121	100.0
196.0577	0.09421	0.00107	100.0
225.5517	0.09130	0.00067	100.0
248.1594	0.10610	0.00204	100.0
275.6798	0.10017	0.00190	100.0

301.4587	0.11872	0.01348	100.0
325.9315	0.08767	0.00094	100.0
350.7789	0.09751	0.00178	100.0
375.2984	0.09632	0.00148	100.0
403.1508	0.09341	0.00112	100.0
436.0787	0.09087	0.00081	100.0
463.4310	0.09323	0.00137	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09682
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08202, 0.11162

NODE TIME OUT INTERVAL = 0.009 SECONDS

25.1189	0.10361	0.00174	100.0
56.6924	0.09065	0.00071	100.0
93.3726	0.09837	0.00120	100.0
121.6888	0.08812	0.00076	100.0
147.5486	0.09881	0.00208	100.0
169.1679	0.10513	0.00177	100.0
196.0497	0.09416	0.00107	100.0
225.5437	0.09173	0.00065	100.0
248.1594	0.10597	0.00192	100.0
275.6798	0.10046	0.00188	100.0
301.4587	0.10920	0.00401	100.0
325.9315	0.08749	0.00094	100.0
350.7789	0.09762	0.00181	100.0
375.2984	0.09609	0.00149	100.0
403.1508	0.09315	0.00108	100.0
436.0787	0.09102	0.00088	100.0
463.4310	0.09360	0.00148	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09639
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08408, 0.10870

NODE TIMEOUT INTERVAL = 0.010

25.1189	0.10367	0.00174	100.0
56.6924	0.09068	0.00071	100.0
93.3726	0.09842	0.00120	100.0
121.6888	0.08843	0.00077	100.0
147.5486	0.09885	0.00209	100.0
169.1679	0.10483	0.00137	100.0
196.0497	0.09394	0.00112	100.0
225.5437	0.09130	0.00068	100.0
248.1594	0.10625	0.00194	100.0
275.6798	0.10007	0.00187	100.0
301.4587	0.12634	0.00920	100.0
325.9315	0.08798	0.00095	100.0
350.7789	0.09727	0.00175	100.0
375.3084	0.09665	0.00152	100.0
403.1508	0.09313	0.00105	100.0
436.0787	0.09089	0.00085	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09733
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.07992, 0.11473

NODE TIMEOUT INTERVAL = 0.011 SECONDS

25.1189	0.10373	0.00174	100.0
56.6924	0.09071	0.00071	100.0
93.3726	0.09847	0.00120	100.0
121.6888	0.08862	0.00078	100.0
147.5486	0.09863	0.00202	100.0
169.1679	0.10627	0.00158	100.0
196.0497	0.09443	0.00106	100.0
225.5437	0.09197	0.00065	100.0
248.1594	0.10615	0.00193	100.0
275.6798	0.10067	0.00189	100.0
301.4587	0.10746	0.00325	100.0
325.9315	0.08803	0.00095	100.0
350.7789	0.09767	0.00174	100.0
375.2984	0.09639	0.00149	100.0
403.1508	0.09416	0.00112	100.0
436.0897	0.09096	0.00083	100.0
463.4310	0.09418	0.00154	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09658
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08467, 0.10849

NODE TIMEOUT INTERVAL = 0.012 SECONDS

25.1189	0.10415	0.00177	100.0
56.6924	0.09074	0.00071	100.0
93.3726	0.09864	0.00120	100.0
121.6888	0.08875	0.00078	100.0
147.5486	0.09873	0.00200	100.0
169.1679	0.10597	0.00165	100.0
196.0497	0.09444	0.00107	100.0
225.5437	0.09152	0.00067	100.0
248.1594	0.10618	0.00195	100.0
275.6798	0.10044	0.00187	100.0
301.4587	0.12081	0.00654	100.0
325.9315	0.08947	0.00143	100.0
350.7789	0.09701	0.00169	100.0
375.2984	0.09671	0.00152	100.0
403.1508	0.09369	0.00113	100.0
436.0787	0.09153	0.00088	100.0
463.4310	0.09380	0.00154	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09740
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08215, 0.11265

NODE TIMEOUT INTERVAL = 0.013 SECONDS

25.1189	0.10430	0.00177	100.0
56.6924	0.09080	0.00071	100.0

93.3726	0.09870	0.00121	100.0
121.6888	0.08881	0.00078	100.0
147.5486	0.09906	0.00214	100.0
169.1679	0.10453	0.00137	100.0
196.0497	0.09397	0.00111	100.0
225.5437	0.09453	0.00074	100.0
248.1594	0.10584	0.00193	100.0
275.6798	0.10023	0.00185	100.0
301.4587	0.11507	0.00419	100.0
325.9315	0.08767	0.00097	100.0
350.7789	0.09769	0.00174	100.0
375.2984	0.09687	0.00150	100.0
403.1508	0.09398	0.00109	100.0
436.0787	0.09165	0.00084	100.0
463.4310	0.09370	0.00143	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09710
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08378, 0.11043

NODE TIMEOUT INTERVAL = 0.014 SECONDS

25.1189	0.10445	0.00177	100.0
56.6924	0.09086	0.00071	100.0
93.3726	0.09876	0.00121	100.0
121.6888	0.08887	0.00078	100.0
147.5486	0.09921	0.00221	100.0
169.1679	0.10493	0.00135	100.0
196.0497	0.09407	0.00107	100.0
225.5437	0.09207	0.00069	100.0
248.1594	0.10423	0.00181	100.0
275.6798	0.10191	0.00217	100.0
301.4587	0.12620	0.00999	100.0
325.9315	0.08754	0.00096	100.0
350.7789	0.09728	0.00176	100.0
375.2984	0.09685	0.00153	100.0
403.1508	0.09429	0.00106	100.0
436.0787	0.09117	0.00081	100.0
463.4430	0.09347	0.00143	100.0
463.4310	0.09351	0.00149	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09764
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08060, 0.11468

NODE TIMEOUT INTERVAL = 0.015

25.1189	0.10448	0.00176	100.0
56.6924	0.09092	0.00071	100.0
93.3726	0.09882	0.00120	100.0
121.6888	0.08896	0.00079	100.0
147.5606	0.09924	0.00216	100.0
169.1679	0.10557	0.00149	100.0
196.0517	0.09331	0.00097	100.0
225.5437	0.09110	0.00062	100.0
248.1594	0.10596	0.00193	100.0

275.6798	0.10042	0.00190	100.0
301.4587	0.11584	0.00770	100.0
325.9315	0.08719	0.00095	100.0
350.7789	0.09856	0.00175	100.0
375.2984	0.09653	0.00148	100.0
403.1508	0.09378	0.00110	100.0
436.1027	0.09247	0.00083	100.0
463.4310	0.09453	0.00166	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09712
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08316, 0.11108

NODE TIMEOUT INTERVAL = 0.025 SECONDS

25.1189	0.10525	0.00180	100.0
56.6924	0.09142	0.00072	100.0
93.3726	0.09910	0.00118	100.0
121.7008	0.08892	0.00071	100.0
147.5486	0.09892	0.00210	100.0
169.1679	0.10600	0.00158	100.0
196.0544	0.09426	0.00106	100.0
225.5437	0.09436	0.00079	100.0
248.1594	0.10668	0.00203	100.0
275.6798	0.10049	0.00189	100.0
301.4642	0.12882	0.00943	100.0
325.9315	0.08905	0.00095	100.0
350.7789	0.09799	0.00186	100.0
375.2984	0.09709	0.00148	100.0
403.1508	0.09514	0.00119	100.0
436.0787	0.09221	0.00087	100.0
463.4310	0.09571	0.00157	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09844
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08063, 0.11624

NODE TIMEOUT INTERVAL = 0.030 SECONDS

25.1189	0.10579	0.00185	100.0
56.6924	0.09160	0.00072	100.0
93.3726	0.09947	0.00118	100.0
121.7128	0.08944	0.00072	100.0
147.5486	0.09887	0.00212	100.0
169.1679	0.10536	0.00123	100.0
196.0497	0.09483	0.00107	100.0
225.5437	0.09412	0.00070	100.0
248.1594	0.10586	0.00195	100.0
275.6798	0.10173	0.00194	100.0
301.4587	0.12302	0.00633	100.0
325.9315	0.09269	0.00103	100.0
350.7789	0.09834	0.00180	100.0
375.2984	0.09616	0.00151	100.0
403.1508	0.09457	0.00113	100.0
436.0787	0.09156	0.00082	100.0
463.4310	0.09568	0.00182	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09835
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08307, 0.11363

NODE TIMEOUT INTERVAL = 0.035 SECONDS

25.1189	0.10609	0.00185	100.0
56.6924	0.09192	0.00074	100.0
93.3726	0.09976	0.00118	100.0
121.7[illegible]28	0.08984	0.00073	100.0
147.5[illegible]16	0.09919	0.00210	100.0
169.1679	0.10840	0.00181	100.0
196.0497	0.09519	0.00112	100.0
225.5437	0.09307	0.00079	100.0
248.1594	0.10403	0.00181	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09857
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08702, 0.11012

NODE TIMEOUT INTERVAL = 0.040 SECONDS

25.1189	0.10676	0.00186	100.0
56.6924	0.09228	0.00075	100.0
93.3726	0.10135	0.00127	100.0
121.7345	0.09074	0.00075	100.0
147.5886	0.09947	0.00204	100.0
169.1679	0.10756	0.00131	100.0
196.0517	0.09529	0.00109	100.0
225.5437	0.09363	0.00075	100.0
248.1594	0.10776	0.00211	100.0
275.6798	0.10262	0.00191	100.0
301.4587	0.12184	0.01459	100.0
325.9315	0.09080	0.00105	100.0
350.7789	0.09945	0.00178	100.0
375.2984	0.09725	0.00146	100.0
403.1508	0.09495	0.00110	100.0
436.0787	0.09111	0.00085	100.0
463.4310	0.09548	0.00161	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09917
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08403, 0.11430

NODE TIMEOUT INTERVAL = 0.045 SECONDS

25.1189	0.10706	0.00187	100.0
56.6924	0.09296	0.00078	100.0
93.3726	0.10223	0.00131	100.0
121.7345	0.09163	0.00077	100.0
147.5486	0.10006	0.00241	100.0
169.1679	0.10684	0.00169	100.0
196.0497	0.09593	0.00114	100.0
225.5437	0.09679	0.00098	100.0
248.1594	0.10477	0.00178	100.0

275.6798	0.10494	0.00216	100.0
301.4587	0.12239	0.00586	100.0
325.9315	0.09285	0.00131	100.0
350.7789	0.10072	0.00186	100.0
375.2984	0.09748	0.00154	100.0
403.1508	0.09570	0.00119	100.0
436.0787	0.09201	0.00087	100.0
463.4310	0.09590	0.00172	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09961
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08498, 0.11424

NODE TIMEOUT INTERVAL = 0.050 SECONDS

25.1189	0.10736	0.00189	100.0
56.6924	0.09435	0.00096	100.0
93.3726	0.10356	0.00141	100.0
121.7705	0.09305	0.00084	100.0
147.5866	0.10102	0.00233	100.0
169.1679	0.10736	0.00159	100.0
196.0477	0.09656	0.00123	100.0
225.5437	0.09418	0.00080	100.0
248.1594	0.10604	0.00185	100.0
275.6798	0.10151	0.00178	100.0
301.4587	0.13242	0.00905	100.0
325.9315	0.09069	0.00098	100.0
350.7789	0.09988	0.00172	100.0
375.2984	0.09986	0.00194	100.0
403.1508	0.09654	0.00131	100.0
436.0787	0.09378	0.00097	100.0
463.4310	0.09892	0.00234	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10053
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08247, 0.11859

NODE TIMEOUT INTERVAL = 0.055 SECONDS

25.1189	0.10766	0.00190	100.0
56.6924	0.09492	0.00101	100.0
93.3726	0.10366	0.00141	100.0
121.7705	0.09382	0.00087	100.0
147.6036	0.10199	0.00254	100.0
169.1679	0.10777	0.00161	100.0
196.0517	0.09459	0.00110	100.0
225.5437	0.09452	0.00083	100.0
248.1594	0.10791	0.00192	100.0
275.6798	0.10215	0.00169	100.0
301.4587	0.11868	0.00941	100.0
325.9625	0.09376	0.00129	100.0
350.7789	0.09899	0.00176	100.0
375.2984	0.09887	0.00155	100.0
403.1508	0.09655	0.00119	100.0
436.1147	0.09570	0.00104	100.0
463.4310	0.09673	0.00168	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10035
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08762, 0.11309

NODE TIMEOUT INTERVAL = 0.060 SECONDS

25.1189	0.10909	0.00203	100.0
56.6924	0.09440	0.00088	100.0
93.3726	0.10449	0.00140	100.0
121.7705	0.09423	0.00083	100.0
147.5486	0.10150	0.00259	100.0
169.1679	0.10915	0.00170	100.0
196.0514	0.09641	0.00113	100.0
225.5437	0.09631	0.00086	100.0
248.1594	0.10638	0.00203	100.0
275.6798	0.10500	0.00202	100.0
301.4587	0.13988	0.01075	100.0
325.9315	0.09154	0.00116	100.0
350.7909	0.10291	0.00219	100.0
375.2984	0.09873	0.00156	100.0
403.1508	0.09620	0.00131	100.0
436.0787	0.09454	0.00088	100.0
463.4310	0.09499	0.00177	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10171
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08075, 0.12267

NODE TIMEOUT INTERVAL = 0.065 SECONDS

25.1189	0.10837	0.00195	100.0
56.6924	0.09771	0.00123	100.0
93.3726	0.10186	0.00118	100.0
121.6785	0.09487	0.00093	100.0
147.6016	0.10040	0.00213	100.0
169.1679	0.10872	0.00159	100.0
196.0617	0.09698	0.00126	100.0
225.5437	0.09856	0.00092	100.0
248.1474	0.10613	0.00200	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10178
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.09267, 0.11089

NODE TIMEOUT INTERVAL = 0.070 SECONDS

25.1189	0.10868	0.00206	100.0
56.6924	0.09742	0.00123	100.0
93.3726	0.10239	0.00123	100.0
121.6785	0.09560	0.00097	100.0
147.5946	0.10070	0.00219	100.0
169.1679	0.10980	0.00170	100.0
196.0497	0.09627	0.00116	100.0
225.5437	0.09618	0.00092	100.0
248.1594	0.10781	0.00209	100.0

275.6798	0.10504	0.00196	100.0
301.4587	0.12438	0.01141	100.0
325.9315	0.09248	0.00138	100.0
350.7789	0.10392	0.00220	100.0
375.2984	0.09964	0.00160	100.0
403.1508	0.09647	0.00125	100.0
436.0787	0.09320	0.00097	100.0
463.4310	0.09593	0.00152	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10133
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08649, 0.11617

NODE TIMEOUT INTERVAL = 0.075 SECONDS

25.1189	0.10925	0.00203	100.0
56.6924	0.09945	0.00142	100.0
93.3726	0.10236	0.00127	100.0
121.6785	0.09655	0.00108	100.0
147.5996	0.10257	0.00250	100.0
169.1679	0.11080	0.00187	100.0
196.0472	0.09732	0.00125	100.0
225.5437	0.09610	0.00102	100.0
248.1594	0.10993	0.00206	100.0
275.6798	0.10756	0.00241	100.0
301.4587	0.11966	0.00882	100.0
325.9315	0.09382	0.00138	100.0
350.7789	0.10085	0.00205	100.0
375.2984	0.10197	0.00207	100.0
403.1868	0.09714	0.00125	100.0
436.0787	0.09799	0.00128	100.0
463.4310	0.10118	0.00265	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10221
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.08928, 0.11513

NODE TIMEOUT INTERVAL = 0.080 SECONDS

25.1189	0.10960	0.00206	100.0
56.6924	0.10019	0.00151	100.0
93.3726	0.10239	0.00128	100.0
121.6785	0.09706	0.00113	100.0
147.6046	0.10282	0.00252	100.0
169.1679	0.11082	0.00178	100.0
196.0497	0.09646	0.00124	100.0
225.5437	0.09844	0.00117	100.0
248.1594	0.11197	0.00250	100.0
275.6798	0.10389	0.00212	100.0
301.4587	0.12213	0.01327	100.0
325.9315	0.09453	0.00144	100.0
350.7789	0.10223	0.00191	100.0
375.2984	0.10230	0.00188	100.0
403.1628	0.10000	0.00132	100.0
436.0787	0.10018	0.00133	100.0
463.4310	0.11034	0.00352	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10351
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.09025, 0.11678

NODE TIMEOUT INTERVAL = 0.085 SECONDS

25.1189	0.11017	0.00212	100.0
56.6924	0.10073	0.00160	100.0
93.3726	0.10434	0.00141	100.0
121.6785	0.09736	0.00130	100.0
147.6096	0.10273	0.00249	100.0
169.1679	0.10922	0.00172	100.0
196.0497	0.09787	0.00138	100.0
225.5437	0.10066	0.00143	100.0
248.1594	0.11090	0.00241	100.0
275.6798	0.10754	0.00206	100.0
301.4587	0.16011	0.02439	100.0
325.9315	0.09526	0.00136	100.0
350.7789	0.10363	0.00226	100.0
375.2984	0.10192	0.00236	100.0
403.1508	0.09753	0.00129	100.0
436.0787	0.09605	0.00116	100.0
463.4310	0.09768	0.00185	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10535
95% CONFIDENCE INTERVAL ON THE ESTIMATE = 0.07773, 0.13297

B. NODE TIMEOUT INTERVAL EXPERIMENT AT HIGH MESSAGE ARRIVAL RATE.

NODE TIMEOUT INTERVAL = 0.0011 SECONDS

23.3323	0.10821	0.00218	100.0
52.6485	0.09646	0.00101	100.0
86.7076	0.10325	0.00139	100.0
113.0029	0.09005	0.00091	100.0
137.0139	0.10315	0.00227	100.0
157.0958	0.11273	0.00258	100.0
182.0584	0.09443	0.00100	100.0
209.4379	0.10250	0.00149	100.0
230.4434	0.12199	0.00569	100.0
255.9929	0.10450	0.00202	100.0
279.9630	0.12409	0.00681	100.0
302.6552	0.08986	0.00103	100.0
325.7293	0.10398	0.00191	100.0
348.4985	0.10202	0.00224	100.0
374.3597	0.09528	0.00128	100.0
404.9347	0.09394	0.00096	100.0
430.3791	0.10327	0.00200	100.0
455.1555	0.09989	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10244
95% CONFIDENCE INTERVAL = 0.08461, 0.12027

NODE TIMEOUT INTERVAL = 0.0012 SECONDS

23.3323	0.10835	0.00219	100.0
52.6498	0.09636	0.00100	100.0
86.7076	0.10319	0.00139	100.0
113.0029	0.08989	0.00092	100.0
137.0139	0.10269	0.00217	100.0
157.0958	0.11279	0.00263	100.0
182.0584	0.09459	0.00100	100.0
209.4379	0.10244	0.00150	100.0
230.4434	0.12191	0.00570	100.0
255.9929	0.10458	0.00201	100.0
279.9630	0.12355	0.00663	100.0
302.6552	0.09007	0.00105	100.0
325.7293	0.10456	0.00210	100.0
348.4997	0.10476	0.00250	100.0
374.3597	0.09538	0.00126	100.0
404.9347	0.09394	0.00096	100.0
430.3803	0.10321	0.00200	100.0
455.1567	0.10008	0.00128	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10259
95% CONFIDENCE INTERVAL = 0.08490, 0.12029

NODE TIMEOUT INTERVAL = 0.0013 SECONDS

23.3323	0.10834	0.00219	100.0
52.6487	0.09664	0.00100	100.0
86.7076	0.10317	0.00140	100.0
113.0029	0.08998	0.00091	100.0
137.0139	0.10266	0.00220	100.0
157.0958	0.11220	0.00246	100.0
182.0584	0.09441	0.00100	100.0
209.4379	0.10255	0.00150	100.0
230.4434	0.12202	0.00569	100.0
255.9929	0.10461	0.00201	100.0
279.9630	0.12868	0.00703	100.0
302.6552	0.09006	0.00106	100.0
325.7293	0.10328	0.00173	100.0
348.4985	0.10183	0.00224	100.0
374.3597	0.09510	0.00125	100.0
404.9347	0.09426	0.00105	100.0
430.3791	0.10326	0.00196	100.0
455.1555	0.09995	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10262
95% CONFIDENCE INTERVAL = 0.08368, 0.12156

NODE TIMEOUT INTERVAL = 0.0014 SECONDS

23.3323	0.10835	0.00219	100.0
52.6488	0.09646	0.00101	100.0
86.7076	0.10319	0.00140	100.0
113.0029	0.09034	0.00098	100.0
137.0139	0.10334	0.00231	100.0

157.0958	0.11261	0.00258	100.0
182.0584	0.09493	0.00100	100.0
209.4379	0.10249	0.00149	100.0
230.4434	0.12203	0.00569	100.0
255.9929	0.10467	0.00201	100.0
279.9630	0.12428	0.00696	100.0
302.6552	0.09002	0.00105	100.0
325.7293	0.10289	0.00170	100.0
348.4985	0.10184	0.00224	100.0
374.3597	0.09509	0.00129	100.0
404.9347	0.09389	0.00096	100.0
430.3791	0.10272	0.00191	100.0
455.1569	0.10002	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10242
95% CONFIDENCE INTERVAL = 0.08465, 0.12020

NODE TIMEOUT INTERVAL = 0.0015 SECONDS

23.3323	0.10873	0.00229	100.0
52.6489	0.09617	0.00100	100.0
86.7076	0.10313	0.00140	100.0
113.0029	0.09000	0.00091	100.0
137.0139	0.10326	0.00233	100.0
157.0958	0.11272	0.00263	100.0
182.0584	0.09695	0.00149	100.0
209.4379	0.10269	0.00152	100.0
230.4434	0.12215	0.00568	100.0
255.9929	0.10465	0.00201	100.0
279.9630	0.14817	0.00917	100.0
302.6552	0.09028	0.00106	100.0
325.7293	0.10494	0.00197	100.0
348.4985	0.10108	0.00218	100.0
374.3597	0.09553	0.00125	100.0
404.9347	0.09424	0.00106	100.0
430.3791	0.10334	0.00201	100.0
455.1555	0.10000	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10394
95% CONFIDENCE INTERVAL = 0.07881, 0.12906

NODE TIMEOUT INTERVAL = 0.0016 SECONDS

23.3323	0.10874	0.00229	100.0
52.6506	0.09623	0.00100	100.0
86.7076	0.10299	0.00139	100.0
113.0029	0.09005	0.00091	100.0
137.0139	0.10303	0.00223	100.0
157.0958	0.11262	0.00259	100.0
182.0584	0.09458	0.00100	100.0
209.4379	0.10237	0.00150	100.0
230.4434	0.12212	0.00571	100.0
255.9929	0.10483	0.00204	100.0
279.9630	0.12349	0.00655	100.0
302.6552	0.09008	0.00105	100.0
325.7293	0.10371	0.00175	100.0

348.5001	0.09934	0.00204	100.0
374.3597	0.09573	0.00129	100.0
404.9347	0.09392	0.00096	100.0
430.3807	0.10343	0.00200	100.0
455.1555	0.10037	0.00129	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10234
95% CONFIDENCE INTERVAL = 0.08462, 0.12006

NODE TIMEOUT INTERVAL = 0.0017 SECONDS

23.3323	0.10877	0.00230	100.0
52.6474	0.09641	0.00100	100.0
86.7076	0.10312	0.00139	100.0
113.0029	0.09006	0.00091	100.0
137.0139	0.10295	0.00217	100.0
157.0958	0.11248	0.00258	100.0
182.0584	0.09462	0.00101	100.0
209.4379	0.10282	0.00151	100.0
230.4434	0.12196	0.00569	100.0
255.9929	0.10465	0.00200	100.0
279.9630	0.12524	0.00748	100.0
302.6552	0.09012	0.00105	100.0
325.7293	0.10329	0.00172	100.0
348.4985	0.09943	0.00205	100.0
374.3597	0.09552	0.00126	100.0
404.9347	0.09431	0.00105	100.0
430.3791	0.10326	0.00200	100.0
455.1555	0.09995	0.00128	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10241
95% CONFIDENCE INTERVAL = 0.08435, 0.12046

NODE TIMEOUT INTERVAL = 0.0018 SECONDS

23.3323	0.10878	0.00230	100.0
52.6474	0.09642	0.00100	100.0
86.7076	0.10313	0.00139	100.0
113.0029	0.09006	0.00091	100.0
137.0139	0.10367	0.00241	100.0
157.0958	0.11283	0.00263	100.0
182.0584	0.09464	0.00102	100.0
209.4379	0.10259	0.00152	100.0
230.4434	0.12212	0.00571	100.0
255.9929	0.10464	0.00201	100.0
279.9630	0.12440	0.00694	100.0
302.6552	0.09019	0.00106	100.0
325.7293	0.10434	0.00183	100.0
348.4985	0.10181	0.00225	100.0
374.3597	0.09539	0.00128	100.0
404.9347	0.09415	0.00097	100.0
430.3791	0.10323	0.00200	100.0
455.1555	0.10003	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10259
95% CONFIDENCE INTERVAL = 0.08470, 0.12049

NODE TIMEOUT INTERVAL = 0.0019 SECONDS

23.3323	0.10879	0.00230	100.0
52.6474	0.09643	0.00100	100.0
86.7076	0.10313	0.00139	100.0
113.0029	0.09007	0.00091	100.0
137.0139	0.10333	0.00227	100.0
157.0958	0.11249	0.00258	100.0
182.0584	0.09445	0.00100	100.0
209.4379	0.10250	0.00149	100.0
230.4434	0.12233	0.00571	100.0
255.9929	0.10462	0.00200	100.0
279.9630	0.12872	0.00821	100.0
302.6552	0.09010	0.00106	100.0
325.7293	0.10347	0.00173	100.0
348.4985	0.09933	0.00204	100.0
374.3597	0.09581	0.00129	100.0
404.9347	0.09414	0.00096	100.0
430.3791	0.10403	0.00204	100.0
455.1555	0.10025	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10268
95% CONFIDENCE INTERVAL = 0.08361, 0.12175

NODE TIMEOUT INTERVAL = 0.0020 SECONDS

23.3323	0.10880	0.00230	100.0
52.6474	0.09645	0.00100	100.0
86.7076	0.10324	0.00139	100.0
113.0029	0.09009	0.00091	100.0
137.0139	0.10371	0.00247	100.0
157.0958	0.11267	0.00259	100.0
182.0584	0.09459	0.00100	100.0
209.4379	0.10257	0.00150	100.0
230.4434	0.12215	0.00570	100.0
255.9929	0.10459	0.00200	100.0
279.9630	0.12884	0.00804	100.0
302.6552	0.09002	0.00105	100.0
325.7293	0.10412	0.00195	100.0
348.4985	0.10185	0.00225	100.0
374.3617	0.09539	0.00128	100.0
404.9347	0.09413	0.00096	100.0
430.3791	0.10347	0.00200	100.0
455.1555	0.10017	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10283
95% CONFIDENCE INTERVAL = 0.08377, 0.12189

NODE TIMEOUT INTERVAL = 0.0021 SECONDS

23.3323	0.10840	0.00219	100.0
52.6474	0.09664	0.00102	100.0
86.7076	0.10297	0.00136	100.0
113.0029	0.09026	0.00091	100.0
137.0139	0.10288	0.00225	100.0

157.0958	0.11294	0.00264	100.0
182.0584	0.09462	0.00101	100.0
209.4379	0.10237	0.00149	100.0
230.4434	0.12204	0.00569	100.0
255.9929	0.10471	0.00200	100.0
279.9630	0.12455	0.00687	100.0
302.6552	0.09027	0.00106	100.0
325.7293	0.10370	0.00161	100.0
348.4985	0.10089	0.00219	100.0
374.3597	0.09628	0.00129	100.0
404.9347	0.09413	0.00098	100.0
430.3791	0.10324	0.00199	100.0
455.1576	0.10001	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10252
95% CONFIDENCE INTERVAL = 0.08473, 0.12030

NODE TIMEOUT INTERVAL = 0.0022 SECONDS

23.3323	0.10841	0.00219	100.0
52.6474	0.09643	0.00100	100.0
86.7076	0.10245	0.00133	100.0
113.0029	0.09044	0.00096	100.0
137.0139	0.10317	0.00227	100.0
157.0958	0.11280	0.00263	100.0
182.0584	0.09460	0.00100	100.0
209.4379	0.10285	0.00154	100.0
230.4434	0.12310	0.00589	100.0
255.9929	0.10477	0.00200	100.0
279.9630	0.12852	0.00765	100.0
302.6552	0.09009	0.00105	100.0
325.7293	0.10342	0.00173	100.0
348.4985	0.10221	0.00225	100.0
374.3619	0.09513	0.00128	100.0
404.9347	0.09558	0.00107	100.0
430.3791	0.10367	0.00201	100.0
455.1555	0.10002	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10286
95% CONFIDENCE INTERVAL = 0.08384, 0.12188

NODE TIMEOUT INTERVAL = 0.0023 SECONDS

23.3323	0.10841	0.00219	100.0
52.6474	0.09644	0.00100	100.0
86.7076	0.10246	0.00133	100.0
113.0029	0.09045	0.00096	100.0
137.0139	0.10366	0.00247	100.0
157.0958	0.11276	0.00263	100.0
182.0584	0.09474	0.00101	100.0
209.4379	0.10280	0.00154	100.0
230.4434	0.12224	0.00569	100.0
255.9929	0.10484	0.00200	100.0
279.9630	0.12448	0.00686	100.0
302.6552	0.09000	0.00105	100.0
325.7293	0.10340	0.00174	100.0

348.4985	0.09952	0.00205	100.0
374.3597	0.09593	0.00132	100.0
404.9347	0.09396	0.00096	100.0
430.3791	0.10387	0.00203	100.0
455.1571	0.10016	0.00128	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10249
95% CONFIDENCE INTERVAL = 0.08462, 0.12036

NODE TIMEOUT INTERVAL = 0.0024 SECONDS

23.3323	0.10842	0.00219	100.0
52.6498	0.09647	0.00100	100.0
86.7076	0.10242	0.00133	100.0
113.0029	0.09012	0.00092	100.0
137.0139	0.10272	0.00215	100.0
157.0958	0.11296	0.00263	100.0
182.0584	0.09451	0.00100	100.0
209.4379	0.10266	0.00151	100.0
230.4434	0.12201	0.00569	100.0
255.9929	0.10458	0.00201	100.0
279.9630	0.12361	0.00662	100.0
302.6552	0.09019	0.00106	100.0
325.7293	0.10457	0.00210	100.0
348.4985	0.10226	0.00228	100.0
374.3597	0.09561	0.00134	100.0
404.9347	0.09405	0.00095	100.0
430.3791	0.10342	0.00203	100.0
455.1555	0.10010	0.00125	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10252
95% CONFIDENCE INTERVAL = 0.08488, 0.12016

NODE TIMEOUT INTERVAL = 0.0025 SECONDS

23.3323	0.10843	0.00219	100.0
52.6474	0.09654	0.00101	100.0
86.7076	0.10292	0.00138	100.0
113.0029	0.08897	0.00074	100.0
137.0139	0.10313	0.00223	100.0
157.0958	0.11275	0.00263	100.0
182.0584	0.09474	0.00100	100.0
209.4379	0.10278	0.00153	100.0
230.4434	0.12208	0.00570	100.0
255.9929	0.10463	0.00201	100.0
279.9630	0.12499	0.00713	100.0
302.6552	0.09005	0.0010[illegible]	100.0
325.7293	0.10302	0.00159	100.0
348.4985	0.10197	0.00224	100.0
374.3597	0.09582	0.00137	100.0
404.9347	0.09443	0.00105	100.0
430.4067	0.16093	0.01504	100.0
455.1571	0.10033	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10553
95% CONFIDENCE INTERVAL = 0.07419, 0.13687

NODE TIMEOUT INTERVAL = 0.026 SECONDS

23.3323	0.10841	0.00219	100.0
52.6500	0.09664	0.00101	100.0
86.7076	0.10326	0.00139	100.0
113.0029	0.09024	0.00092	100.0
137.0139	0.10254	0.00216	100.0
157.0958	0.11279	0.00265	100.0
182.0584	0.09473	0.00100	100.0
209.4379	0.10331	0.00160	100.0
230.4434	0.12211	0.00576	100.0
255.9929	0.10486	0.00201	100.0
279.9630	0.12451	0.00710	100.0
302.6552	0.09006	0.00105	100.0
325.7293	0.10331	0.00170	100.0
348.5011	0.10258	0.00231	100.0
374.3597	0.09494	0.00122	100.0
404.9373	0.09414	0.00105	100.0
430.3791	0.10330	0.00200	100.0
455.1555	0.09966	0.00125	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10254
95% CONFIDENCE INTERVAL = 0.08464, 0.12044

NODE TIMEOUT INTERVAL = 0.0027 SECONDS

23.3323	0.10841	0.00219	100.0
52.6474	0.09660	0.00102	100.0
86.7076	0.10324	0.00139	100.0
113.0029	0.09028	0.00091	100.0
137.0166	0.10365	0.00240	100.0
157.0958	0.11289	0.00259	100.0
182.0584	0.09466	0.00100	100.0
209.4379	0.10284	0.00154	100.0
230.4434	0.12232	0.00567	100.0
255.9929	0.10471	0.00202	100.0
279.9986	0.12532	0.00756	100.0
302.6552	0.09012	0.00107	100.0
325.7293	0.10357	0.00174	100.0
348.4985	0.10205	0.00225	100.0
374.3597	0.09541	0.00130	100.0
404.9347	0.09403	0.00097	100.0
430.3818	0.10334	0.00199	100.0
455.1555	0.10005	0.00128	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10265
95% CONFIDENCE INTERVAL = 0.08453, 0.12077

NODE TIMEOUT INTERVAL = 0.0028 SECONDS

23.3323	0.10842	0.00219	100.0
52.6474	0.09635	0.00101	100.0
86.7076	0.10324	0.00139	100.0
113.0029	0.09029	0.00091	100.0
137.0139	0.10378	0.00245	100.0

348.4985	0.09931	0.00204	100.0
374.3597	0.09647	0.00140	100.0
404.9347	0.09408	0.00097	100.0
430.4807	0.14099	0.00616	100.0
455.1555	0.10020	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10463
95% CONFIDENCE INTERVAL = 0.07972, 0.12954

NODE TIMEOUT INTERVAL = 0.0031 SECONDS

23.3323	0.10862	0.00219	100.0
52.6474	0.09637	0.00100	100.0
86.7076	0.10245	0.00133	100.0
113.0029	0.09040	0.00099	100.0
137.0139	0.10339	0.00231	100.0
157.0958	0.11254	0.00258	100.0
182.0584	0.09475	0.00101	100.0
209.4379	0.10274	0.00149	100.0
230.4434	0.12224	0.00570	100.0
255.9929	0.10473	0.00199	100.0
279.9630	0.12710	0.00754	100.0
302.6552	0.09026	0.00107	100.0
325.7293	0.10360	0.00172	100.0
348.4985	0.10212	0.00225	100.0
374.3597	0.09558	0.00128	100.0
404.9347	0.09415	0.00095	100.0
430.3791	0.13513	0.00640	100.0
455.1555	0.09992	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10436
95% CONFIDENCE INTERVAL = 0.08103, 0.12768

NODE TIMEOUT INTERVAL = 0.0032 SECONDS

23.3323	0.10848	0.00219	100.0
52.6474	0.09640	0.00101	100.0
86.7076	0.10253	0.00134	100.0
113.0029	0.09024	0.00091	100.0
137.0139	0.10401	0.00251	100.0
157.0958	0.11289	0.00264	100.0
182.0584	0.09490	0.00103	100.0
209.4379	0.10263	0.00149	100.0
230.4434	0.12236	0.00569	100.0
255.9929	0.10469	0.00200	100.0
279.9630	0.12395	0.00659	100.0
302.6552	0.09009	0.00099	100.0
325.7293	0.10342	0.00170	100.0
348.4985	0.09958	0.00204	100.0
374.3597	0.09596	0.00130	100.0
404.9347	0.09561	0.00114	100.0
430.3791	0.13301	0.00547	100.0
455.1555	0.09999	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10409
95% CONFIDENCE INTERVAL = 0.08196, 0.12623

NODE TIMEOUT INTERVAL = 0.0033 SECONDS

23.3323	0.10848	0.00219	100.0
52.6474	0.09640	0.00101	100.0
86.7076	0.10253	0.00134	100.0
113.0029	0.09025	0.00091	100.0
137.0139	0.10321	0.00220	100.0
157.0958	0.11280	0.00263	100.0
182.0584	0.09505	0.00096	100.0
209.4379	0.10267	0.00149	100.0
230.4434	0.12219	0.00569	100.0
255.9929	0.10509	0.00204	100.0
279.9630	0.12399	0.00678	100.0
302.6552	0.09036	0.00107	100.0
325.7293	0.10380	0.00162	100.0
348.4985	0.10113	0.00218	100.0
374.3597	0.09608	0.00128	100.0
404.9347	0.09421	0.00097	100.0
430.3791	0.10340	0.00200	100.0
455.1555	0.10014	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10257
95% CONFIDENCE INTERVAL = 0.08492, 0.12020

NODE TIMEOUT INTERVAL = 0.0034 SECONDS

23.3323	0.10853	0.00219	100.0
52.6474	0.09655	0.00102	100.0
86.7076	0.10324	0.00140	100.0
113.0029	0.09033	0.00091	100.0
137.0139	0.10308	0.00225	100.0
157.0958	0.11281	0.00265	100.0
182.0584	0.09720	0.00140	100.0
209.4379	0.10106	0.00149	100.0
230.4434	0.12210	0.00571	100.0
255.9929	0.10492	0.00204	100.0
279.9630	0.12942	0.00836	100.0
302.6586	0.09027	0.00106	100.0
325.7293	0.10390	0.00161	100.0
348.4985	0.10112	0.00219	100.0
374.3597	0.09540	0.00123	100.0
404.9347	0.09433	0.00104	100.0
430.3791	0.10343	0.00199	100.0
455.1571	0.10015	0.00129	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10289
95% CONFIDENCE INTERVAL = 0.08393, 0.12184

NODE TIMEOUT INTERVAL = 0.0035 SECONDS

23.3323	0.10853	0.00219	100.0
52.6474	0.09656	0.00102	100.0
86.7076	0.10324	0.00140	100.0
113.0029	0.09034	0.00091	100.0
137.0139	0.10380	0.00257	100.0

157.0958	0.11278	0.00263	100.0
182.0584	0.09721	0.00140	100.0
209.4379	0.10106	0.00149	100.0
230.4434	0.12211	0.00571	100.0
255.9929	0.10493	0.00204	100.0
279.9630	0.12765	0.00777	100.0
302.6552	0.09028	0.00106	100.0
325.7293	0.10385	0.00173	100.0
348.4985	0.09950	0.00204	100.0
374.3597	0.09551	0.00118	100.0
404.9347	0.09418	0.00103	100.0
430.3791	0.13294	0.00508	100.0
455.1555	0.10003	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10429
95% CONFIDENCE INTERVAL = 0.08154, 0.12704

IODE TIMEOUT INTERVAL = 0.0036 SECONDS

23.3323	0.10854	0.00219	100.0
52.6474	0.09656	0.00102	100.0
86.7076	0.10325	0.00140	100.0
113.0029	0.09035	0.00091	100.0
137.0139	0.10297	0.00221	100.0
157.0958	0.11278	0.00263	100.0
182.0584	0.09721	0.00140	100.0
209.4379	0.10107	0.00149	100.0
230.4434	0.12211	0.00571	100.0
255.9929	0.10495	0.00204	100.0
279.9630	0.12458	0.00700	100.0
302.6588	0.09036	0.00106	100.0
325.7293	0.10360	0.00174	100.0
348.4985	0.10229	0.00225	100.0
374.3597	0.09498	0.00121	100.0
404.9347	0.09407	0.00097	100.0
430.3791	0.10351	0.00200	100.0
455.1555	0.09941	0.00120	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10261
95% CONFIDENCE INTERVAL = 0.08486, 0.12032

NODE TIMEOUT INTERVAL = 0.0037 SECONDS

23.3323	0.10855	0.00219	100.0
52.6474	0.09656	0.00102	100.0
86.7076	0.10325	0.00140	100.0
113.0029	0.09036	0.00091	100.0
137.0139	0.10309	0.00225	100.0
157.0958	0.11305	0.00262	100.0
182.0584	0.09462	0.00100	100.0
209.4379	0.10255	0.00149	100.0
230.4434	0.12207	0.00569	100.0
255.9929	0.10465	0.00200	100.0
279.9630	0.12391	0.00654	100.0
302.6552	0.08986	0.00098	100.0
325.7293	0.10429	0.00161	100.0

348.4985	0.10101	0.00218	100.0
374.3597	0.09549	0.00124	100.0
404.9347	0.09407	0.00096	100.0
430.3791	0.10347	0.00199	100.0
455.1555	0.10011	0.00128	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10257
95% CONFIDENCE INTERVAL = 0.08486, 0.12029

NODE TIMEOUT INTERVAL = 0.0038 SECONDS

23.3323	0.10855	0.00220	100.0
52.6474	0.09657	0.00102	100.0
86.7076	0.10325	0.00140	100.0
113.0029	0.09037	0.00091	100.0
137.0139	0.10325	0.00227	100.0
157.0958	0.11234	0.00247	100.0
182.0584	0.09463	0.00100	100.0
209.4417	0.10095	0.00149	100.0
230.4434	0.11571	0.00390	100.0
255.9929	0.10473	0.00201	100.0
279.9630	0.12560	0.00775	100.0
302.6552	0.09033	0.00105	100.0
325.7293	0.10378	0.00157	100.0
348.4985	0.10319	0.00237	100.0
374.3597	0.09573	0.00126	100.0
404.9347	0.09425	0.00096	100.0
430.3791	0.10430	0.00202	100.0
455.1555	0.10036	0.00131	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10235
95% CONFIDENCE INTERVAL = 0.08553, 0.11917

NODE TIMEOUT INTERVAL = 0.0039 SECONDS

23.3323	0.10856	0.00220	100.0
52.6474	0.09657	0.00102	100.0
86.7076	0.10326	0.00140	100.0
113.0029	0.09038	0.00091	100.0
137.0139	0.10346	0.00236	100.0
157.0958	0.11258	0.00245	100.0
182.0584	0.09464	0.00100	100.0
209.4418	0.10257	0.00150	100.0
230.4434	0.12217	0.00568	100.0
255.9929	0.10477	0.00200	100.0
279.9630	0.12725	0.00758	100.0
302.6552	0.08988	0.00099	100.0
325.7293	0.10431	0.00166	100.0
348.4985	0.10357	0.00238	100.0
374.3597	0.09548	0.00120	100.0
404.9347	0.09418	0.00097	100.0
430.3791	0.10445	0.00213	100.0
455.1571	0.10034	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10307
95% CONFIDENCE INTERVAL = 0.08464, 0.12150

NODE TIMEOUT INTERVAL = 0.004 SECONDS

23.3323	0.10856	0.00220	100.0
52.6474	0.09658	0.00102	100.0
86.7076	0.10318	0.00139	100.0
113.0029	0.09037	0.00091	100.0
137.0139	0.10419	0.00266	100.0
157.0958	0.11283	0.00257	100.0
182.0584	0.09484	0.00108	100.0
209.4379	0.10293	0.00155	100.0
230.4434	0.12244	0.00570	100.0
255.9929	0.10495	0.00205	100.0
279.9630	0.12762	0.00730	100.0
302.6552	0.09023	0.00106	100.0
325.7293	0.10515	0.00199	100.0
348.4985	0.10393	0.00241	100.0
374.3597	0.09589	0.00133	100.0
404.9387	0.09423	0.00096	100.0
430.3791	0.10303	0.00191	100.0
455.1555	0.10032	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10307
95% CONFIDENCE INTERVAL = 0.08440, 0.12173

NODE TIMEOUT INTERVAL = 0.005 SECONDS

23.3323	0.10863	0.00219	100.0
52.6474	0.09677	0.00102	100.0
86.7076	0.10325	0.00139	100.0
113.0029	0.09072	0.00091	100.0
137.0139	0.10383	0.00234	100.0
157.0958	0.11322	0.00265	100.0
182.0584	0.09492	0.00100	100.0
209.4379	0.10100	0.00149	100.0
230.4434	0.12260	0.00566	100.0
255.9929	0.10463	0.00203	100.0
279.9630	0.12569	0.00773	100.0
302.6552	0.09027	0.00105	100.0
325.7293	0.10399	0.00163	100.0
348.4985	0.10064	0.00213	100.0
374.3597	0.09621	0.00125	100.0
404.9347	0.09509	0.00109	100.0
430.3791	0.10335	0.00201	100.0
455.1555	0.10004	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10273
95% CONFIDENCE INTERVAL = 0.08465, 0.12082

NODE TIMEOUT INTERVAL = 0.006 SECONDS

23.3423	0.10928	0.00237	100.0
52.6474	0.09665	0.00100	100.0
86.7076	0.10328	0.00138	100.0
113.0029	0.08924	0.00074	100.0
137.0199	0.10295	0.00212	100.0

157.0958	0.11294	0.00263	100.0
182.0602	0.09503	0.00101	100.0
209.4379	0.10275	0.00152	100.0
230.4434	0.12202	0.00570	100.0
255.9929	0.10515	0.00200	100.0
279.9630	0.12668	0.00748	100.0
302.6552	0.09016	0.00105	100.0
325.7293	0.10490	0.00199	100.0
348.4985	0.10223	0.00224	100.0
374.3597	0.09485	0.00122	100.0
404.9347	0.09576	0.00108	100.0
430.3791	0.13164	0.00467	100.0
455.1555	0.10002	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10434
95% CONFIDENCE INTERVAL = 0.08196, 0.12672

NODE TIMEOUT INTERVAL = 0.007 SECONDS

23.3423	0.10926	0.00237	100.0
52.6474	0.09666	0.00104	100.0
86.7076	0.10182	0.00125	100.0
113.0029	0.09049	0.00093	100.0
137.0139	0.10402	0.00226	100.0
157.0958	0.11312	0.00261	100.0
182.0584	0.09565	0.00103	100.0
209.4379	0.10322	0.00150	100.0
230.4434	0.12270	0.00567	100.0
255.9929	0.10490	0.00207	100.0
279.9630	0.12916	0.00754	100.0
302.6552	0.09076	0.00107	100.0
325.7293	0.10328	0.00170	100.0
348.4985	0.09990	0.00206	100.0
374.3597	0.09587	0.00130	100.0
404.9417	0.09463	0.00104	100.0
430.3791	0.14097	0.00702	100.0
455.1555	0.10074	0.00129	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10496
95% CONFIDENCE INTERVAL = 0.07969, 0.13024

NODE TIMEOUT INTERVAL = 0.008 SECONDS

23.3423	0.10975	0.00239	100.0
52.6474	0.09689	0.00106	100.0
86.7076	0.10368	0.00137	100.0
113.0029	0.08987	0.00084	100.0
137.0139	0.10339	0.00224	100.0
157.0958	0.11271	0.00257	100.0
182.0584	0.09805	0.00152	100.0
209.4379	0.10332	0.00150	100.0
230.4434	0.12278	0.00567	100.0
255.9929	0.10495	0.00208	100.0
279.9630	0.12861	0.00708	100.0
302.6552	0.09087	0.00106	100.0
325.7293	0.10290	0.00157	100.0

348.4985	0.10207	0.00223	100.0
374.3597	0.09537	0.00122	100.0
404.9347	0.09630	0.00109	100.0
430.4537	0.13929	0.00600	100.0
455.1555	0.09985	0.00127	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10514
95% CONFIDENCE INTERVAL = 0.08072, 0.12957

NODE TIMEOUT INTERVAL = 0.009 SECONDS

23.3323	0.10954	0.00226	100.0
52.6474	0.09689	0.00100	100.0
86.7076	0.10360	0.00137	100.0
113.0029	0.09105	0.00093	100.0
137.0139	0.10442	0.00254	100.0
157.0958	0.11346	0.00256	100.0
182.0584	0.09486	0.00100	100.0
209.4469	0.10501	0.00162	100.0
230.4434	0.12228	0.00565	100.0
255.9929	0.10489	0.00202	100.0
279.9630	0.12493	0.00666	100.0
302.6552	0.09076	0.00105	100.0
325.7293	0.10413	0.00177	100.0
348.4985	0.10495	0.00247	100.0
374.3597	0.09638	0.00128	100.0
404.9347	0.09508	0.00100	100.0
430.3691	0.12704	0.00608	100.0
455.1555	0.10000	0.00119	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10458
95% CONFIDENCE INTERVAL = 0.08403, 0.12513

NODE TIMEOUT INTERVAL = 0.010 SECONDS

23.3323	0.10941	0.00223	100.0
52.6474	0.09698	0.00102	100.0
86.7076	0.10212	0.00125	100.0
113.0029	0.09076	0.00093	100.0
137.0139	0.10478	0.00240	100.0
157.0958	0.11310	0.00262	100.0
182.0584	0.09496	0.00100	100.0
209.4479	0.10282	0.00146	100.0
230.4434	0.12223	0.00568	100.0
255.9929	0.10491	0.00201	100.0
279.9630	0.12542	0.00771	100.0
302.6552	0.09073	0.00107	100.0
325.7293	0.10383	0.00166	100.0
348.4985	0.10334	0.00244	100.0
374.3597	0.09619	0.00125	100.0
404.9347	0.09593	0.00110	100.0
430.4391	0.14347	0.00753	100.0
455.1555	0.10026	0.00126	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10520
95% CONFIDENCE INTERVAL = 0.08014, 0.13025

NODE TIMEOUT INTERVAL = 0.011 SECONDS

23.3423	0.10989	0.00238	100.0
52.6474	0.09647	0.00104	100.0
86.7076	0.10293	0.00133	100.0
113.0029	0.09094	0.00091	100.0
137.0139	0.10468	0.00226	100.0
157.0958	0.11319	0.00262	100.0
182.0584	0.09494	0.00100	100.0
209.4489	0.10505	0.00162	100.0
230.4434	0.12230	0.00565	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10454
95% CONFIDENCE INTERVAL = 0.08740, 0.12167

NODE TIMEOUT INTERVAL = 0.012 SECONDS

23.3423	0.11043	0.00238	100.0
52.6474	0.09652	0.00104	100.0
86.7076	0.10321	0.00134	100.0
113.0029	0.09090	0.00091	100.0
137.0139	0.10409	0.00210	100.0
157.0958	0.11296	0.00261	100.0
182.0584	0.09838	0.00145	100.0
209.4379	0.10152	0.00147	100.0
230.4434	0.12241	0.00566	100.0
256.0049	0.10460	0.00200	100.0
279.9986	0.12902	0.00708	100.0
302.6552	0.09135	0.00105	100.0
325.7293	0.10596	0.00213	100.0
348.4985	0.10453	0.00250	100.0
374.3597	0.09561	0.00121	100.0
404.9347	0.09413	0.00097	100.0
430.3911	0.13641	0.00617	100.0
455.1555	0.10074	0.00123	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10546
95% CONFIDENCE INTERVAL = 0.08204, 0.12889

NODE TIMEOUT INTERVAL = 0.013 SECONDS

23.3323	0.10959	0.00223	100.0
52.6474	0.09633	0.00100	100.0
86.7076	0.10325	0.00134	100.0
113.0029	0.09092	0.00091	100.0
137.0149	0.10475	0.00225	100.0
157.0958	0.11347	0.00255	100.0
182.0584	0.09868	0.00152	100.0
209.4379	0.10302	0.00148	100.0
230.4434	0.12259	0.00580	100.0
256.0049	0.10508	0.00193	100.0
279.9630	0.12421	0.00666	100.0
302.6552	0.09167	0.00105	100.0
325.7293	0.10585	0.00178	100.0
348.4985	0.10167	0.00223	100.0

374.3597	0.09640	0.00125	100.0
404.9347	0.09758	0.00111	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10951
95% CONFIDENCE INTERVAL = 0.06329, 0.15572

NODE TIMEOUT INTERVAL = 0.014 SECONDS

23.3323	0.10994	0.00224	100.0
52.6474	0.09603	0.00100	100.0
86.7076	0.10216	0.00119	100.0
113.0029	0.09173	0.00095	100.0
137.0279	0.10494	0.00266	100.0
157.0958	0.11489	0.00278	100.0
182.0724	0.09813	0.00149	100.0
209.4499	0.10211	0.00158	100.0
230.4434	0.12226	0.00584	100.0
255.9929	0.10541	0.00201	100.0
279.9630	0.13648	0.00775	100.0
302.6552	0.09199	0.00107	100.0
325.7293	0.10548	0.00186	100.0
348.4985	0.10243	0.00224	100.0
374.3597	0.09763	0.00134	100.0
404.9347	0.09559	0.00103	100.0
430.4077	0.15788	0.01348	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10757
95% CONFIDENCE INTERVAL = 0.07576, 0.13938

NODE TIMEOUT INTERVAL = 0.015 SECONDS

23.3323	0.10985	0.00224	100.0
52.6474	0.09604	0.00100	100.0
86.7076	0.10219	0.00119	100.0
113.0029	0.09177	0.00095	100.0
137.0139	0.10484	0.00271	100.0
157.0958	0.11418	0.00259	100.0
182.0584	0.09915	0.00146	100.0
209.4379	0.10164	0.00148	100.0
230.4434	0.11645	0.00385	100.0
255.9929	0.10593	0.00210	100.0
279.9630	0.12839	0.00776	100.0
302.6672	0.09216	0.00103	100.0
325.7293	0.10221	0.00172	100.0
348.4985	0.10298	0.00235	100.0
374.3597	0.09549	0.00111	100.0
404.9347	0.09745	0.00120	100.0
430.3791	0.13703	0.00608	100.0
455.1555	0.10161	0.00134	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10510
95% CONFIDENCE INTERVAL = 0.08255, 0.12764

NODE TIMEOUT INTERVAL = 0.020 SECONDS

23.3323	0.11017	0.00226	100.0

52.6474	0.09644	0.00100	100.0
86.7076	0.10179	0.00123	100.0
113.0029	0.08979	0.00079	100.0
137.0139	0.10515	0.00227	100.0
157.0958	0.11760	0.00337	100.0
182.0584	0.09759	0.00124	100.0
209.4379	0.10337	0.00151	100.0
230.4452	0.12409	0.00572	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10508
95% CONFIDENCE INTERVAL = 0.08625, 0.12390

NODE TIMEOUT INTERVAL = 0.025 SECONDS

23.3323	0.11026	0.00228	100.0
52.6594	0.09791	0.00114	100.0
86.7076	0.10274	0.00123	100.0
113.0409	0.09275	0.00126	100.0
137.0269	0.10845	0.00368	100.0
157.1101	0.11026	0.00180	100.0
182.0733	0.09868	0.00128	100.0
209.4379	0.09830	0.00084	100.0
230.4322	0.11139	0.00244	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10386
95% CONFIDENCE INTERVAL = 0.09156, 0.11616

NODE TIMEOUT INTERVAL = 0.03 SECONDS

23.3323	0.11089	0.00243	100.0
52.6474	0.09997	0.00165	100.0
86.7076	0.10194	0.00129	100.0
113.0409	0.09465	0.00136	100.0
137.0439	0.10938	0.00368	100.0
157.1382	0.11314	0.00218	100.0
182.0733	0.09918	0.00126	100.0
209.4379	0.09685	0.00077	100.0
230.4342	0.11103	0.00238	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10408
95% CONFIDENCE INTERVAL = 0.09181, 0.11634

NODE TIMEOUT INTERVAL = 0.035 SECONDS

23.3323	0.11032	0.00234	100.0
52.6594	0.09876	0.00119	100.0
86.7076	0.10554	0.00136	100.0
113.0409	0.09365	0.00119	100.0
137.0259	0.10958	0.00368	100.0
157.1101	0.11134	0.00200	100.0
182.0733	0.09784	0.00122	100.0
209.4379	0.09883	0.00086	100.0
230.4342	0.11073	0.00244	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10405
95% CONFIDENCE INTERVAL = 0.09209, 0.11600

C. HOST TIMEOUT INTERVAL EXPERIMENT DATA

HOST TIMEOUT = 0.30 SECONDS

23.3267	0.13316	0.01039	100.0
52.6474	0.10643	0.00380	100.0
86.7076	0.10101	0.00209	100.0
113.0029	0.08671	0.00092	100.0
137.0159	0.09397	0.00182	100.0
157.1142	0.10726	0.00154	100.0
182.0526	0.09140	0.00155	100.0
209.4379	0.09444	0.00086	100.0
230.4434	0.10683	0.00229	100.0
255.9929	0.09323	0.00098	100.0
279.9250	0.12709	0.01107	100.0
302.6552	0.09002	0.00132	100.0
325.7173	0.09310	0.00145	100.0
348.4985	0.09368	0.00129	100.0
374.3717	0.09034	0.00081	100.0
404.9347	0.08901	0.00074	100.0
430.3628	0.11202	0.00528	100.0
455.1330	0.08514	0.00055	100.0
476.7992	0.09457	0.00163	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09944
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07429, 0.12460

HOST TIMEOUT = 0.32 SECONDS

23.3047	0.13660	0.01087	100.0
52.6594	0.08628	0.00051	100.0
86.7076	0.09382	0.00103	100.0
113.0509	0.08377	0.00053	100.0
137.0139	0.09925	0.00314	100.0
158.0287	0.20531	0.05759	100.0
182.0355	0.16567	0.03492	100.0
209.4379	0.09149	0.00142	100.0
230.4202	0.10930	0.00284	100.0
255.9929	0.09718	0.00210	100.0
279.9250	0.15146	0.01927	100.0
302.6552	0.08585	0.00074	100.0
325.7293	0.09972	0.00366	100.0
348.4985	0.09301	0.00112	100.0
374.3717	0.09154	0.00169	100.0
404.9347	0.09355	0.00110	100.0
430.3333	0.13640	0.01203	100.0
455.1330	0.08777	0.00069	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.11155
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.04707, 0.17604

HOST TIMEOUT = 0.33 SECONDS

23.3649	0.12401	0.00828	100.0
52.6594	0.11446	0.01080	100.0
86.7096	0.11056	0.01021	100.0
112.9924	0.08453	0.00078	100.0
137.0139	0.09245	0.00203	100.0
157.0922	0.11954	0.00523	100.0
182.0526	0.09238	0.00154	100.0
209.4379	0.09634	0.00339	100.0
230.4322	0.10391	0.00233	100.0
255.9929	0.09850	0.00221	100.0
279.8890	0.19568	0.04987	100.0
302.6552	0.08574	0.00084	100.0
325.7173	0.10150	0.00308	100.0
348.4985	0.09230	0.00102	100.0
374.3717	0.08995	0.00093	100.0
404.9347	0.09442	0.00164	100.0
430.3333	0.14408	0.01693	100.0
455.1330	0.08964	0.00151	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10666
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.05652, 0.15680

HOST TIMEOUT = 0.34 SECONDS

23.3283	0.15317	0.02140	100.0
52.6594	0.08903	0.00070	100.0
86.7096	0.09362	0.00107	100.0
112.9924	0.09400	0.00366	100.0
137.0139	0.09580	0.00254	100.0
157.0912	0.18576	0.04074	100.0
182.0372	0.08810	0.00071	100.0
209.4379	0.08974	0.00068	100.0
230.4174	0.12079	0.00598	100.0
255.9929	0.09450	0.00156	100.0
279.9740	0.15940	0.01609	100.0
302.6572	0.08796	0.00092	100.0
325.7293	0.09371	0.00129	100.0
348.4985	0.09219	0.00126	100.0
374.3717	0.08700	0.00078	100.0
404.9347	0.09594	0.00183	100.0
430.3333	0.08992	0.00208	100.0
455.1330	0.08444	0.00054	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10460
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.04942, 0.15979

HOST TIMEOUT = 0.36 SECONDS

23.3283	0.15317	0.02140	100.0
52.6594	0.08903	0.00070	100.0
86.7096	0.09362	0.00107	100.0
112.9924	0.09400	0.00366	100.0
137.0139	0.09580	0.00254	100.0
157.0912	0.18576	0.04074	100.0

182.0372	0.08810	0.00071	100.0
209.4379	0.08974	0.00068	100.0
230.4174	0.12079	0.00598	100.0
255.9929	0.09450	0.00156	100.0
279.9740	0.15940	0.01609	100.0
302.6572	0.08796	0.00092	100.0
325.7293	0.09371	0.00129	100.0
348.4985	0.09219	0.00126	100.0
374.3717	0.08700	0.00078	100.0
404.9347	0.09594	0.00183	100.0
430.3333	0.08992	0.00208	100.0
455.1330	0.08444	0.00054	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10460
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.04942, 0.15979

HOST TIMEOUT = 0.38 SECONDS

23.3283	0.13186	0.01121	100.0
52.6474	0.10469	0.01125	100.0
86.7076	0.09275	0.00088	100.0
112.9924	0.08634	0.00061	100.0
137.0139	0.09554	0.00286	100.0
157.0858	0.09699	0.00127	100.0
182.0355	0.09883	0.00519	100.0
209.4379	0.09228	0.00184	100.0
230.4322	0.10227	0.00194	100.0
255.9929	0.09930	0.00257	100.0
279.9630	0.11843	0.00723	100.0
302.6552	0.08725	0.00103	100.0
325.7173	0.08717	0.00052	100.0
348.4985	0.09437	0.00127	100.0
374.3717	0.08526	0.00082	100.0
404.9347	0.08818	0.00067	100.0
430.3353	0.09819	0.00348	100.0
455.1330	0.08716	0.00073	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09826
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07387, 0.12265

HOST TIMEOUT = 0.40 SECONDS

23.4349	0.13941	0.01444	100.0
52.6474	0.08919	0.00078	100.0
86.7096	0.10355	0.00536	100.0
112.9924	0.08384	0.00067	100.0
137.0139	0.09898	0.00320	100.0
157.0861	0.10052	0.00195	100.0
182.0584	0.09456	0.00313	100.0
209.4379	0.09403	0.00200	100.0
230.4174	0.10344	0.00227	100.0
255.9929	0.09352	0.00104	100.0
279.9980	0.11771	0.00695	100.0
302.6552	0.09207	0.00239	100.0

325.7173	0.10752	0.00610	100.0
348.4985	0.09120	0.00101	100.0
374.3717	0.08849	0.00080	100.0
404.9347	0.09388	0.00208	100.0
430.3333	0.09019	0.00236	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09838
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07341, 0.12335

HOST TIMEOUT = 0.41 SECONDS

23.3283	0.21821	0.08201	100.0
52.6594	0.12600	0.03058	100.0
86.7076	0.09904	0.00403	100.0
112.9924	0.08667	0.00082	100.0
137.0139	0.09680	0.00202	100.0
157.1032	0.12008	0.00579	100.0
182.0546	0.09072	0.00097	100.0
209.4379	0.09946	0.00259	100.0
230.4174	0.11330	0.00594	100.0
255.9929	0.09585	0.00234	100.0
279.8890	0.13377	0.00995	100.0
302.6552	0.08894	0.00225	100.0
325.7173	0.11010	0.00995	100.0
348.4985	0.09467	0.00302	100.0
374.3717	0.08939	0.00097	100.0
404.9347	0.08980	0.00067	100.0
430.3333	0.08837	0.00214	100.0
455.1330	0.08712	0.00070	100.0
476.7992	0.10183	0.00422	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10685
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.04901, 0.16469

HOST TIMEOUT = 0.42 SECONDS

23.3007	0.15170	0.02406	100.0
52.6594	0.08981	0.00092	100.0
86.7076	0.09213	0.00088	100.0
112.9924	0.08402	0.00053	100.0
137.0159	0.09792	0.00241	100.0
157.1022	0.10133	0.00132	100.0
182.0526	0.08699	0.00059	100.0
209.4379	0.08968	0.00073	100.0
230.4174	0.09865	0.00158	100.0
255.9929	0.09401	0.00215	100.0
279.9510	0.11737	0.00786	100.0
302.6552	0.09761	0.00609	100.0
325.7293	0.08657	0.00044	100.0
348.4985	0.09214	0.00113	100.0
374.3717	0.08894	0.00097	100.0
404.9347	0.09417	0.00235	100.0
430.3333	0.08808	0.00111	100.0
455.1330	0.08513	0.00053	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09810
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.06575, 0.13045

HOST TIMEOUT = 0.43 SECONDS

23.3283	0.15007	0.02458	100.0
52.6594	0.08668	0.00066	100.0
86.7076	0.09535	0.00111	100.0
113.0284	0.08719	0.00099	100.0
137.0139	0.09684	0.00243	100.0
157.0861	0.13498	0.02307	100.0
182.0526	0.08874	0.00079	100.0
209.4379	0.10294	0.00439	100.0
230.4342	0.10747	0.00298	100.0
255.9929	0.09107	0.00082	100.0
279.9740	0.12669	0.00955	100.0
302.6552	0.09526	0.00407	100.0
325.7293	0.08741	0.00056	100.0
348.5005	0.09167	0.00095	100.0
374.3737	0.08932	0.00099	100.0
404.9347	0.09475	0.00280	100.0
430.3333	0.08949	0.00265	100.0
455.1332	0.08891	0.00086	100.0
476.7992	0.27224	0.10511	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10932
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.02674, 0.19190

HOST TIMEOUT = 0.44 SECONDS

23.3283	0.10187	0.00132	100.0
52.6614	0.08748	0.00069	100.0
86.7076	0.09788	0.00148	100.0
113.0044	0.08575	0.00103	100.0
137.0139	0.09413	0.00181	100.0
157.0958	0.11031	0.00268	100.0
182.0355	0.08804	0.00064	100.0
209.4379	0.08905	0.00076	100.0
230.4184	0.10306	0.00199	100.0
255.9929	0.09252	0.00139	100.0
279.9740	0.11335	0.00470	100.0
302.6552	0.08763	0.00115	100.0
325.7293	0.08865	0.00058	100.0
348.5005	0.09190	0.00121	100.0
374.3717	0.08996	0.00089	100.0
404.9347	0.08678	0.00073	100.0
430.3333	0.08507	0.00100	100.0
455.1330	0.08751	0.00069	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09320
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07735, 0.10904

HOST TIMEOUT INTERVAL = 0.45

23.3283	0.12163	0.00840	100.0
52.6474	0.09280	0.00274	100.0
86.7076	0.09245	0.00096	100.0
113.0284	0.08501	0.00071	100.0
137.0159	0.09280	0.00155	100.0
157.1022	0.12721	0.01198	100.0
182.1975	0.12677	0.02465	100.0
209.4379	0.10033	0.00431	100.0
230.4322	0.10075	0.00172	100.0
255.9949	0.11051	0.00687	100.0
279.8890	0.13659	0.03105	100.0
302.6552	0.09379	0.00285	100.0
325.7293	0.08811	0.00056	100.0
348.4985	0.09369	0.00107	100.0
374.3717	0.09228	0.00284	100.0
404.9347	0.09729	0.00399	100.0
430.3333	0.09410	0.00299	100.0
455.1330	0.08784	0.00071	100.0
476.7992	0.13449	0.03077	100.0
476.7992	0.08981	0.00101	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10360
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07126, 0.13594

HOST TIMEOUT = 0.46 SECONDS

23.3283	0.17924	0.04442	100.0
52.6594	0.12719	0.03349	100.0
86.7076	0.09462	0.00097	100.0
112.9924	0.08907	0.00269	100.0
137.0139	0.10491	0.01124	100.0
157.1262	0.10357	0.00159	100.0
182.0566	0.09169	0.00086	100.0
209.4379	0.09987	0.00496	100.0
230.4322	0.09856	0.00161	100.0
255.9929	0.09993	0.00347	100.0
279.9250	0.11154	0.00743	100.0
302.6552	0.08923	0.00267	100.0
325.7293	0.11097	0.01188	100.0
348.4985	0.08990	0.00097	100.0
374.3597	0.08674	0.00077	100.0
404.9347	0.11511	0.01193	100.0
430.3333	0.09003	0.00270	100.0
455.1330	0.08868	0.00080	100.0
476.7992	0.09186	0.00273	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10330
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.06249, 0.14411

HOST TIMEOUT = 0.47 SECONDS

23.3283	0.13836	0.01604	100.0

52.6594	0.09295	0.00263	100.0
86.7076	0.09917	0.00378	100.0
113.0284	0.09095	0.00275	100.0
137.0139	0.09276	0.00163	100.0
157.0861	0.10852	0.00448	100.0
182.0426	0.09738	0.00447	100.0
209.4379	0.09989	0.00316	100.0
230.4174	0.11033	0.00609	100.0
255.9949	0.09263	0.00103	100.0
279.9740	0.10929	0.00434	100.0
302.6552	0.09256	0.00313	100.0
325.7173	0.09598	0.00439	100.0
348.4985	0.09406	0.00128	100.0
374.3717	0.08984	0.00092	100.0
404.9347	0.08894	0.00074	100.0
430.3333	0.10003	0.00534	100.0
455.1330	0.08883	0.00082	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10049
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07524, 0.12573

HOST TIMEOUT = 0.48 SECONDS

23.3112	0.13881	0.02474	100.0
52.6594	0.09545	0.00343	100.0
86.7076	0.11074	0.01457	100.0
112.9924	0.08237	0.00068	100.0
137.0139	0.10847	0.01370	100.0
157.1078	0.09839	0.00101	100.0
182.0536	0.09452	0.00372	100.0
209.4379	0.08745	0.00071	100.0
230.4434	0.12145	0.00975	100.0
255.9929	0.09134	0.00121	100.0
279.9380	0.11309	0.00559	100.0
302.6552	0.08848	0.00130	100.0
325.7293	0.09197	0.00247	100.0
348.4985	0.09221	0.00124	100.0
374.3717	0.09044	0.00104	100.0
404.9347	0.08929	0.00071	100.0
430.3333	0.08822	0.00117	100.0
455.1330	0.08834	0.00087	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10565
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.03948, 0.17183

HOST TIMEOUT INTERVAL = 0.50 SECONDS

23.3112	0.13570	0.01398	100.0
52.6474	0.08606	0.00051	100.0
86.7076	0.12524	0.02647	100.0
112.9924	0.08606	0.00131	100.0
137.0139	0.09452	0.00193	100.0
157.1022	0.09845	0.00077	100.0
182.0355	0.11272	0.01113	100.0

209.4379	0.08793	0.00055	100.0
230.4174	0.10308	0.00178	100.0
255.9929	0.10507	0.00694	100.0
279.9740	0.16479	0.03915	100.0
302.6552	0.08781	0.00099	100.0
325.7293	0.10066	0.00522	100.0
348.4985	0.09158	0.00106	100.0
374.3597	0.08750	0.00077	100.0
404.9347	0.08814	0.00070	100.0
430.3333	0.09048	0.00361	100.0
455.1369	0.08987	0.00351	100.0
476.7992	0.09327	0.00136	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10152
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.06236, 0.14068

HOST TIMEOUT = 0.51 SECONDS

23.3112	0.13974	0.02117	100.0
52.6474	0.08616	0.00051	100.0
86.7076	0.11087	0.01490	100.0
113.0044	0.08645	0.00071	100.0
137.0139	0.09793	0.00268	100.0
157.1101	0.10133	0.00115	100.0
182.1975	0.08753	0.00059	100.0
209.4379	0.09879	0.00486	100.0
230.4322	0.10061	0.00153	100.0
255.9949	0.10013	0.00400	100.0
279.9380	0.11257	0.00739	100.0
302.6552	0.08689	0.00090	100.0
325.7293	0.09214	0.00104	100.0
348.4985	0.09398	0.00127	100.0
374.3597	0.09447	0.00313	100.0
404.9347	0.09553	0.00317	100.0
430.3333	0.09178	0.00320	100.0
455.1330	0.08876	0.00084	100.0
476.7992	0.12384	0.01602	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09945
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.07296, 0.12594

HOST TIMEOUT = 0.54 SECONDS

23.3283	0.14714	0.02663	100.0
52.6474	0.08725	0.00064	100.0
86.7076	0.09241	0.00096	100.0
113.0284	0.08433	0.00065	100.0
137.0402	0.11155	0.01836	100.0
157.0922	0.10535	0.00445	100.0
182.0526	0.09378	0.00366	100.0
209.4379	0.09276	0.00359	100.0
230.4322	0.10138	0.00209	100.0
255.9929	0.09682	0.00174	100.0
279.9250	0.09751	0.00187	100.0

302.6552	0.09208	0.00358	100.0
325.7173	0.12332	0.01666	100.0
348.4985	0.09264	0.00116	100.0
374.3597	0.09454	0.00338	100.0
404.9347	0.13799	0.04239	100.0
430.3333	0.10233	0.00685	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10158
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.06877, 0.13439

HOST TIMEOUT INTERVAL = 0.55 SECONDS

23.3063	0.19752	0.08749	100.0
52.6594	0.08656	0.00062	100.0
86.7096	0.11864	0.02080	100.0
112.9924	0.08653	0.00103	100.0
137.0484	0.14015	0.04689	100.0
157.1698	0.10407	0.00400	100.0
182.0355	0.09158	0.00327	100.0
209.4379	0.11134	0.00940	100.0
230.4174	0.10299	0.00209	100.0
255.9929	0.09931	0.00214	100.0
279.9980	0.10667	0.00440	100.0
302.6552	0.09732	0.00726	100.0
325.7173	0.14311	0.04495	100.0
348.4985	0.09641	0.00346	100.0
374.3477	0.09045	0.00085	100.0
404.9347	0.08860	0.00077	100.0
430.3333	0.09541	0.00406	100.0
455.1330	0.09627	0.00339	100.0
455.1330	0.08678	0.00079	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10850
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.05595, 0.16104

HOST TIMEOUT = 0.56 SECONDS

23.3283	0.15140	0.02436	100.0
52.6594	0.21370	0.21568	100.0
86.7076	0.09474	0.00103	100.0
113.0409	0.08444	0.00062	100.0
137.0139	0.09979	0.00615	100.0
157.1022	0.09923	0.00119	100.0
182.0392	0.10384	0.01013	100.0
209.4379	0.09644	0.00404	100.0
230.4174	0.10499	0.00227	100.0
255.9929	0.10539	0.00739	100.0
279.9390	0.11322	0.00831	100.0
302.6552	0.09073	0.00363	100.0
325.7293	0.27653	0.24149	100.0
348.5005	0.13359	0.02885	100.0
374.3717	0.09028	0.00096	100.0
404.9347	0.09082	0.00078	100.0
430.6260	0.11382	0.01542	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.13036
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.01172, 0.24901

HOST TIMEOUT = 0.57 SECONDS

23.3183	0.20621	0.10964	100.0
52.6474	0.08663	0.00065	100.0
86.7076	0.09230	0.00095	100.0
113.0029	0.08998	0.00388	100.0
137.2770	0.12492	0.04995	100.0
157.0978	0.24633	0.13616	100.0
182.0296	0.10521	0.00986	100.0
209.4379	0.08664	0.00057	100.0
230.4322	0.09853	0.00170	100.0
255.9929	0.11515	0.01177	100.0
279.9250	0.11886	0.01178	100.0
302.6552	0.09132	0.00410	100.0
325.7293	0.09263	0.00343	100.0
348.4985	0.09089	0.00112	100.0
374.3477	0.09513	0.00393	100.0
404.9347	0.08811	0.00075	100.0
430.3333	0.09852	0.00710	100.0
455.1330	0.10562	0.00933	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.12220
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.01092, 0.23349

HOST TIMEOUT = 0.58 SECONDS

23.3283	0.10187	0.00132	100.0
52.6614	0.08748	0.00069	100.0
86.7076	0.09788	0.00148	100.0
113.0044	0.08575	0.00103	100.0
137.0139	0.09413	0.00181	100.0
157.1022	0.09998	0.00164	100.0
182.0355	0.08905	0.00068	100.0
209.4379	0.09151	0.00081	100.0
230.4174	0.10517	0.00254	100.0
255.9929	0.08996	0.00118	100.0
280.0350	0.10161	0.00300	100.0
302.6552	0.08894	0.00146	100.0
325.7293	0.09052	0.00079	100.0
348.4985	0.09113	0.00106	100.0
374.3717	0.08982	0.00104	100.0
404.9347	0.09022	0.00084	100.0
430.3333	0.08960	0.00125	100.0
455.1330	0.09155	0.00091	100.0
476.7992	0.08965	0.00113	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.09294
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.08236, 0.10352

HOST TIMEOUT = 0.59 SECONDS

23.3047	0.34010	0.26695	100.0
52.6614	0.08318	0.00041	100.0
86.7076	0.09388	0.00090	100.0
113.0509	0.08627	0.00102	100.0
137.0139	0.13857	0.05272	100.0
157.0838	0.09547	0.00094	100.0
182.0296	0.09172	0.00392	100.0
209.4379	0.09004	0.00068	100.0
230.4322	0.10042	0.00155	100.0
255.9929	0.10386	0.00500	100.0
279.8890	0.18640	0.09152	100.0
302.6552	0.09580	0.00416	100.0
325.7173	0.14961	0.06993	100.0
348.4985	0.09031	0.00091	100.0
374.3717	0.08738	0.00074	100.0
404.9935	0.18865	0.04888	100.0
430.3333	0.09442	0.00392	100.0
455.1330	0.08681	0.00075	100.0
455.1350	0.08643	0.00066	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.12789
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.00114, 0.25463

HOST TIMEOUT INTERVAL = 0.60 SECONDS

23.3303	0.18356	0.05304	100.0
52.6474	0.10075	0.00871	100.0
86.7076	0.10568	0.00893	100.0
112.9924	0.09255	0.00388	100.0
137.0139	0.06990	0.00189	100.0
157.1228	0.14926	0.03236	100.0
182.0426	0.08716	0.00077	100.0
209.4399	0.08896	0.00065	100.0
230.4174	0.10137	0.00181	100.0
255.9929	0.09889	0.00476	100.0
279.9740	0.11391	0.00821	100.0
302.6552	0.09086	0.00410	100.0
325.7293	0.09979	0.00764	100.0
348.4985	0.09558	0.00148	100.0
374.3717	0.08779	0.00097	100.0
404.9347	0.11446	0.01445	100.0
430.3333	0.17038	0.10667	100.0
455.1350	0.08610	0.00065	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.10784
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.05354, 0.16215

HOST TIMEOUT = 0.605 SECONDS

23.3047	0.43447	0.54789	100.0
52.6474	0.08725	0.00058	100.0
86.7076	0.09579	0.00088	100.0

113.0029	0.09148	0.00423	100.0
137.0139	0.09919	0.00285	100.0
157.1022	0.10795	0.00478	100.0
182.0426	0.10633	0.01223	100.0
209.4379	0.08797	0.00060	100.0
230.4092	0.10302	0.00203	100.0
255.9929	0.15779	0.04989	100.0
279.9740	0.10802	0.00652	100.0
302.6552	0.15266	0.06184	100.0
325.7173	0.09047	0.00080	100.0
348.5005	0.08996	0.00098	100.0
374.3717	0.08960	0.00101	100.0
404.9347	0.09165	0.00097	100.0
430.3333	0.10996	0.01240	100.0

ESTIMATE OF MEAN RESPONSE TIME = 0.12374
95% CONFIDENCE INTERVAL ON ESTIMATE = 0.0, 0.16215

APPENDIX D. BOX PLOTS OF HOST TIMEOUT EXPERIMENT RUNS

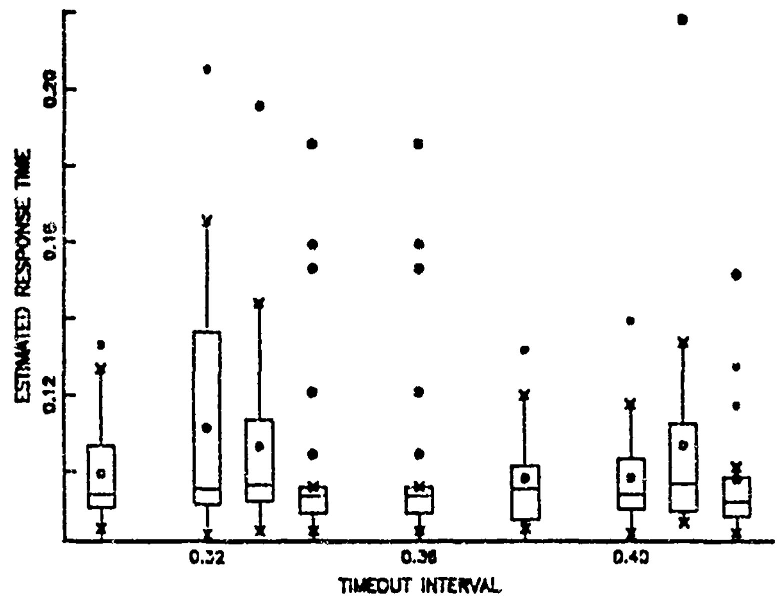

Figure 22. Box Plot of A Portion of the Host Timeout Experiment Runs

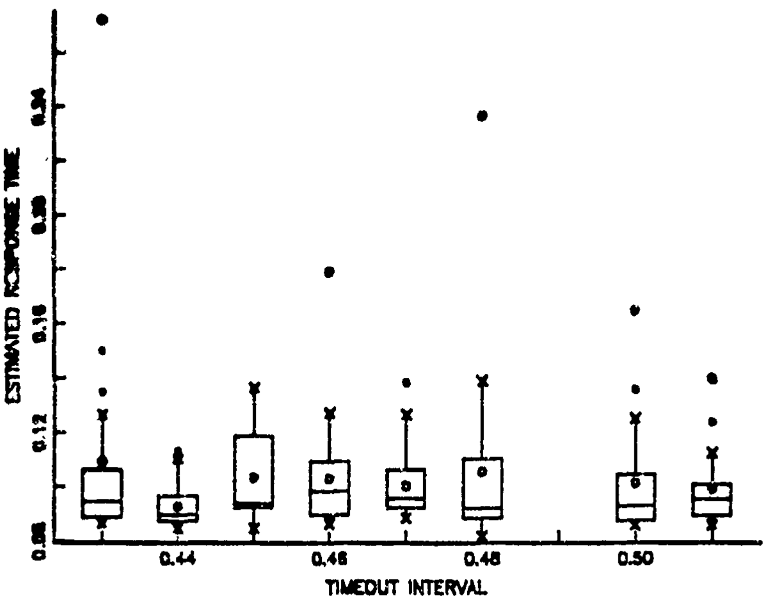

Figure 23. Box Plot of A Portion of the Host Timeout Experiment Runs

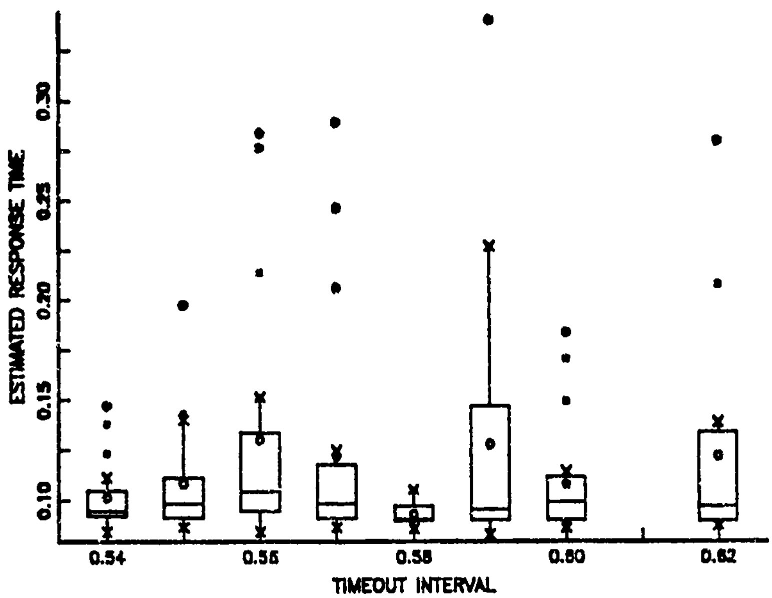

Figure 24. Box Plot of A Portion of the Host Timeout Experiment Runs

LIST OF REFERENCES

1. BBN Communications Corp., *Defense Data Network System Description (Final)*, DCA Circular 310-P70-1, Defense Communications Agency, 1987.

2. Tanenbaum, Andrew S., *Computer Networks*, Prentice Hall Inc., 1981.

3. DeVere, R., and others, *The DDN (Defense Data Network) Course*, Network Strategies Inc., 1986.

4. Feinter, Elizabeth, and others, *DDN (Defense Data Network) Protocol Handbook*, SRI International Inc., 1985.

5. BBN Communications Corp. Report 6874, *Computer Program Functional Specification, Packet Switching Node (PSN)*, 1988.

6. Postel, Jonathan B., Sunshine, Carl A., and Cohen, Danny, *The ARPA Internet Protocol*, IEEE *Computer Networks*, Volume 5, Number 4, 1981.

7. Pritsker, A. Alan B., *Introduction to Simulation and SLAM II*, Systems Publishing Corp., 1986.

8. Cohn, S., and others, *Congestion Control Study Final Report*, BBN Communications Corp. Report 5943, 1985.

9. Zinky, J., Khanna, A., and Vichniac, Gerald, *Performance of the New Routing Metric for ARPANET and MILNET*, BBN Communicaton Corp., 1989.

10. Law, Averill M., and Kelton, W. David, *Simulation Modeling and Analysis*, McGraw-Hill Book Co., 1982.

11. Welch, Peter D., *Computer Performance Modeling Handbook, Chapter 6*, Academic Press, Inc., 1983.